Deep One-Class Learning

A Deep Learning Approach to Anomaly Detection

Deep One-Class Learning

A Deep Learning Approach to Anomaly Detection

vorgelegt von

Lukas Ruff, M.Sc.

an der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin

zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
– Dr. rer. nat. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitz:	Prof. Dr. Benjamin Blankertz
Gutachter:	Prof. Dr. Klaus-Robert Müller
	Prof. Dr. Marius Kloft
	Prof. Dr. Sergios Theodoridis

Tag der wissenschaftlichen Aussprache: 30. April 2021

Berlin 2021

To my wife and my parents

Acknowledgments

There are many people to whom I would like to express my deepest gratitude. I feel humbled and consider myself immensely fortunate to have had the opportunity to meet and work with so many brilliant and wonderful people on my journey.

The seeds of some of the results presented in this thesis have been sowed in the summer of 2017 in Berlin. At that time, I was still in my master's studies and lucky to get involved in early and vivid discussions on deep anomaly detection together with Marius Kloft, Robert Vandermeulen, Stephan Mandt, Lucas Deecke, Nico Görnitz, and Klaus-Robert Müller. Your infectious enthusiasm made me lastingly excited about the topic and have inspired me to begin my research journey.

My heartfelt gratitude go to Klaus and Marius for being exceptional doctoral advisors from the start. You are both mentors and role models to me, and I am deeply grateful to you for sharing your insights, wisdom, and curiosity. I am especially grateful to Klaus for emphasizing scientific insight and solving interdisciplinary problems with machine learning in his research and group, which creates such a diverse and stimulating learning environment. I am grateful to Marius in particular for sharing his mathematical insight with me in many of our long discussions, regularly challenging our views and intuition. Working with you both has been an invaluable and continuous source of inspiration for me.

Moreover, I am grateful to Sergios Theodoridis for accepting my invitation and taking the time to serve as an external supervisor. I am very honored to have such a renowned and distinguished researcher as a member of my doctoral committee.

My special thanks also goes to Emmanuel Müller for showing me a data mining perspective on many challenging problems. I am thankful to have had the opportunity to spend my first research months in the KDD group at HPI, before the group moved to Bonn and I joined the ML group at TU Berlin.

Furthermore, I owe my special thanks to Rob (Robert), who has been immensely patient with my earliest paper drafts and allowed me to practice and improve one edit at a time. You have taught me scientific writing, to say the least, in addition to sharing your technical expertise in our many discussions. This typo is for yuo.

I am also deeply thankful to all the exceptional people I have had the opportunity to work and collaborate with: Alex Binder, Penny Chong, Lucas Deecke, Thomas Dietterich, Billy Joe Franks, Nico Görnitz, Michael Joswig, Marek Kaluba, Jacob Kauffmann, Stephan Mandt, Grégoire Montavon, Wojciech Samek, Thomas Schnake, Shoaib Siddiqui, and Yuri Zemlyanskiy. Thank you each and every one of you.

Many thanks also to all the students, whom I was pleased to supervise in their studies: Julius Coburger, Gabriel Dernbach, Gregor von Dulong, Willi Gierke, Philipp Liznerski, Matteo Paltenghi, René Saitenmacher, Jan Seipp, Dennis Wagner, and Jannik Wolff. I hope you have learned at least as much as I have learned from you in our discussions.

Also, a big thank you to all my other lab mates for sharing the fun! At TU Berlin: Maximilian Alber, Christopher Anders, Mihail Bogojeski, Stephanie Brandl, Thanh Binh Bui, Stefan Chmiela, Sergej Dogadov, Oliver Eberle, Malte Esders, Michael Gastegger, Niklas Gebauer, Miriam Hägele, Adrian Hill, Marina Höhne, Pan Kessel, David Lassner, Jonas Lederer, Simon Letzgus, Lorenz Linhardt, Shinichi Nakajima, Danny Panknin, Huziel Sauceda, Florian Schulz, Kristof Schütt, Philipp Seegerer, Guido Schwenk, Lorenz Vaitl, Ludwig Winkler, and Andreas Ziehe, with special thanks to Cecilia Bonetti and Andrea Gerdes as well as Dominik Kühne for the organizational and technical support respectively. At HPI: Fabian Geier, Thomas Görttler, Davide Mottin, Erik Scharwächter, Arvind Shekar, and Anton Tsitsulin. Thank you for the fun lunch and coffee breaks, and some memorable Christmas Karaoke sessions.

In addition, I acknowledge financial support from the German Federal Ministry of Transport and Digital Infrastructure (BMVI) in the project OSIMAB (FKZ: 19F2017E) during my first year of studies, and by the German Federal Ministry of Education and Research (BMBF) in the project ALICE III (01IS18049B).

Lastly, I owe my dearest gratitude to my wife Nicole and my family for their loving patience and their continued support in every conceivable way.

Abstract

Anomaly detection is the problem of identifying unusual patterns in data. This problem is relevant for a wide variety of applications in various domains such as fault and damage detection in manufacturing, fraud detection in finance and insurance, intrusion detection in cybersecurity, disease detection in medical diagnosis, or scientific discovery. Many of these applications involve increasingly complex data at large scale, for instance, large collections of images or text. The lack of effective solutions in such settings has sparked an interest in developing anomaly detection methods based on deep learning, which has enabled breakthroughs in other machine learning problems that involve large amounts of complex data.

This thesis proposes *Deep One-Class Learning*, a deep learning approach to anomaly detection that is based on the one-class classification paradigm. One-class classification views anomaly detection from a classification perspective, aiming to learn a discriminative decision boundary that separates the normal from the anomalous data. In contrast to previous methods that rely on fixed (usually manually engineered) features, deep one-class learning expands the one-class classification approach with methods that learn (or transfer) data representations via suitable one-class learning objectives. The key idea underlying deep one-class learning is to learn a transformation (e.g., a deep neural network) in such a way that the normal data points are concentrated in feature space, causing anomalies to deviate from the concentrated region, thereby making them detectable.

We introduce several deep one-class learning methods in this thesis that follow the above idea while integrating different assumptions about the data or a specific domain. These include semi-supervised variants that can incorporate labeled anomalies, for example, or specific methods for images and text that enable model interpretability and an explanation of anomalies. Moreover, we present a unifying view of anomaly detection methods that, in addition to one-class classification, also covers reconstruction methods as well as methods based on density estimation and generative modeling. For each of these main approaches, we identify connections between respective deep and "shallow" methods based on common underlying principles.

Through multiple experiments and analyses, we demonstrate that deep one-class learning is useful for anomaly detection, especially on semantic detection tasks. Finally, we conclude this thesis by discussing limits of the proposed approach and outlining specific paths for future research.

Zusammenfassung

„Anomalieerkennung“ beschreibt die Problemstellung ungewöhnliche Muster in Daten zu erkennen. Dieses Problem ist für verschiedenste Anwendungen relevant, beispielsweise zur Fehler- und Schadenserkennung in der Industrie, zur Ermittlung von Finanz- oder Versicherungsbetrug, dem Aufspüren von Cyberattacken, der Feststellung von Krankheiten in der Medizin oder um wissenschaftliche Entdeckungen datengestützt voranzutreiben. Viele dieser Anwendungen umfassen zunehmend komplexe Daten in großem Umfang, etwa große Mengen an hochauflösenden Bildern oder Text. Der Mangel effektiver Lösungen in derartig komplexen Anwendungen hat das Interesse an der Entwicklung von Methoden zur Anomalieerkennung basierend auf *Deep Learning* geweckt, womit bereits Durchbrüche in anderen Problemstellungen, die große Mengen komplexer Daten umfassen, erzielt werden konnten.

In dieser Dissertation wird ein neuer *Deep Learning* Ansatz zur Anomalieerkennung vorgeschlagen, *Deep One-Class Learning*, der auf dem Prinzip der Ein-Klassen-Klassifikation beruht. Die Einklassen-Klassifikation interpretiert das Problem der Anomalieerkennung als ein einseitiges Klassifikationsproblem und zielt darauf ab, eine diskriminierende Entscheidungsgrenze zu lernen, die die normalen von den anomalen Daten trennt.

Im Gegensatz zu bisherigen Methoden, die auf fixierten (meist manuell konstruierten) Datenmerkmalen beruhen, erweitert *Deep One-Class Learning* den Ein-Klassen-Klassifikationsansatz um Methoden, die relevante Datenrepräsentationen mittels geeigneter Ein-Klassen-Lernziele aus den Daten lernen (oder übertragen) können. Die Kernidee von *Deep One-Class Learning* besteht darin, eine Transformation der Daten (z.B. ein tiefes neuronales Netzwerk) so zu optimieren, dass diese die normalen Datenpunkte im Merkmalsraum konzentriert, wodurch sich Anomalien vom konzentrierten Bereich im Merkmalsraum abheben und dadurch erkennbar werden.

Im Rahmen dieser Dissertation werden mehrere *Deep One-Class Learning* Methoden eingeführt, die genau dieser Kernidee folgen, und dabei unterschiedliche Annahmen an die Daten oder ein bestimmtes Anwendungsgebiet integrieren. Dazu gehören beispielsweise halbüberwachte Varianten, die bekannte Anomalien in das Lernen eines Modells miteinbeziehen können, oder spezifische Methoden für Bild- und Textdaten, die eine Modellinterpretierbarkeit sowie eine Erklärung der Anomalien ermöglichen. Des Weiteren wird in dieser Arbeit eine vereinheitlichende Betrachtung von Methoden zur Anomalieerkennung vorgestellt, die neben Methoden der Ein-Klassen-Klassifikation auch Rekonstruktionsmethoden sowie Methoden zur

Schätzung von Datenverteilungen umfasst. Auf Basis gemeiner Grundprinzipien dieser Hauptansätze, ermöglicht diese Betrachtung Verbindungen zwischen den jeweiligen *Deep Learning* und klassischen Methoden herzustellen.

Anhand mehrerer Experimente und Analysen wird gezeigt, dass *Deep One-Class Learning* ein nützlicher Ansatz zur Anomalieerkennung ist, insbesondere zur Erkennung semantischer Anomalien. Die Arbeit schließt mit einer Diskussion der Grenzen des vorgeschlagenen Ansatzes sowie einer Skizzierung offener Probleme und zukünftiger Forschung ab.

Contents

1 Introduction and Overview **1**
- 1.1 Motivation . . . 2
  - 1.1.1 Why Is Anomaly Detection Relevant? . . . 2
  - 1.1.2 Why Deep Learning for Anomaly Detection? . . . 2
- 1.2 The Thesis . . . 3
  - 1.2.1 Contributions and Findings . . . 3
  - 1.2.2 List of Publications . . . 5
  - 1.2.3 Organization of the Thesis . . . 6
- 1.3 Formal Introduction to Anomaly Detection . . . 7
  - 1.3.1 Problem Definition . . . 7
  - 1.3.2 Data Settings and Properties . . . 13
  - 1.3.3 Evaluation . . . 17
  - 1.3.4 Challenges . . . 19

2 One-Class Learning **21**
- 2.1 Shallow One-Class Classification . . . 21
  - 2.1.1 One-Class vs. Binary Classification . . . 21
  - 2.1.2 One-Class Classification in Input Space . . . 23
  - 2.1.3 Kernel-Based One-Class Classification . . . 24
- 2.2 Deep One-Class Classification . . . 25
  - 2.2.1 The Deep SVDD Method . . . 25
  - 2.2.2 Theoretical Properties of Deep SVDD . . . 27
  - 2.2.3 Regularization and Variants . . . 30
  - 2.2.4 Experimental Evaluation on MNIST and CIFAR-10 . . . 31
- 2.3 Deep Semi-Supervised One-Class Classification . . . 35
  - 2.3.1 Negative Examples . . . 36
  - 2.3.2 The Deep SAD Method . . . 37
  - 2.3.3 An Information-Theoretic View . . . 38
  - 2.3.4 Experimental Evaluation on Using Few True Anomalies . . . 40
  - 2.3.5 Hypersphere Classification . . . 46
  - 2.3.6 Experimental Evaluation on Using Many Auxiliary Anomalies . . . 47

3 Applications to Computer Vision and NLP **51**
3.1 Explainable One-Class Classification for Images . . . 51
3.1.1 The FCDD Method . . . 52
3.1.2 Experimental Evaluation . . . 55
3.2 Multi-Context One-Class Classification for Text . . . 61
3.2.1 The CVDD Method . . . 62
3.2.2 Experimental Evaluation . . . 66

4 A Unifying View of Anomaly Detection **73**
4.1 Probabilistic Methods . . . 74
4.1.1 Classic Density Estimation . . . 74
4.1.2 Deep Generative Models . . . 74
4.1.3 Energy-Based Models . . . 77
4.2 Reconstruction Methods . . . 78
4.2.1 Reconstruction Objective . . . 79
4.2.2 Principal Component Analysis . . . 81
4.2.3 Autoencoders . . . 82
4.2.4 Clustering Models . . . 84
4.3 Unifying View . . . 85
4.3.1 Modeling Dimensions of Anomaly Detection Methods . . . 86
4.3.2 Comparative Discussion . . . 87
4.3.3 Distance-based Anomaly Detection . . . 89
4.4 Comparative Evaluation . . . 89
4.4.1 Experimental Evaluation on MNIST-C and MVTec-AD . . . 90
4.4.2 The "Clever Hans" Effect in Anomaly Detection . . . 91

5 Conclusion and Outlook **97**
5.1 Conclusion . . . 97
5.2 Future Research Paths . . . 98
5.2.1 Unexplored Combinations of Modeling Dimensions . . . 98
5.2.2 Bridging Related Lines of Research on Robustness . . . 99
5.2.3 Interpretability and Trustworthiness . . . 101
5.2.4 The Need for Challenging and Open Datasets . . . 102
5.2.5 Weak Supervision and Self-Supervised Learning . . . 103
5.2.6 Foundation and Theory . . . 104

Appendix **109**

A Ablation Studies and Sensitivity Analyses **109**
A.1 Deep SAD Embedding Dimensionality Sensitivity Analysis . . . 109
A.2 Hypersphere Classifier Ablation Study . . . 109
A.3 FCDD Receptive Field Sensitivity Analysis . . . 110
A.4 FCDD Gaussian Upsampling Sensitivity Analysis . . . 111

B Supplementary Details **113**
B.1 Details of Experimental Evaluation on Using Few True Anomalies . . . 113
B.1.1 Network Architectures . . . 113
B.1.2 Training Details of Competing Methods . . . 113
B.2 FCDD Network Architectures . . . 115
B.3 Training Details of Experiments on MNIST-C and MVTec-AD . . . 117

C Supplementary Results **119**
C.1 Best vs. Second Best on CIFAR-10 when Using Few True Anomalies . . 119
C.2 Full Results of Experimental Evaluation on Using Few True Anomalies . 121
C.3 FCDD Results on Individual Classes . . . 125
C.4 Average Precision on MNIST-C and MVTec-AD . . . 127

Bibliography **129**

1 Introduction and Overview

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. *Anomaly detection* (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. Well-known methods for anomaly detection include Principal Component Analysis (PCA) [420, 238, 494, 234, 244], the One-Class Support Vector Machine (OC-SVM) [496], Support Vector Data Description (SVDD) [538], nearest neighbor algorithms [283, 441, 78], and Kernel Density Estimation (KDE) [460, 417].

What the above methods have in common is that they are all *unsupervised*, which constitutes the main approach to anomaly detection. This is because labeled anomalous data is often non-existent in standard anomaly detection settings. If such data is available, it is usually insufficient to fully characterize "anomalousness" due to the heterogeneity of anomalies: anything not normal is, by definition, an anomaly. This makes a supervised approach typically ineffective. Because of this, the standard approach to anomaly detection is to learn a model of normality from normal data in an unsupervised manner, so that anomalies become detectable through deviations from the model.

Finding the relevant data features, the signal among the noise, is key to the success of any machine learning task—including anomaly detection. Classic anomaly detection methods such as the ones above can require extensive feature engineering to be effective. For these "shallow" methods, feature engineering and training a model typically constitute two distinct steps. Deep learning [307, 491, 187], on the other hand, combines these two steps. Using multi-layered neural networks, the aim of deep learning is to learn a model and a relevant data representation *jointly*. Deep learning has enabled breakthroughs in many domains [307, 491], but these successes have initially been limited to mainly supervised learning tasks. This is because defining a useful unsupervised representation learning objective is inherently difficult [50].

This thesis proposes a deep learning approach to anomaly detection that is based on the one-class classification paradigm. The approach, which we call *Deep One-Class Learning*, follows the idea of learning a transformation (e.g., a deep neural network) such that the normal data gets concentrated feature space, causing anomalies to be mapped away from the concentrated region, thereby making them detectable.

1.1 Motivation

1.1.1 Why Is Anomaly Detection Relevant?

The study of anomaly detection 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 [150], the problem of anomaly detection has likely been studied informally even earlier, since anomalies are phenomena that naturally occur in diverse academic disciplines such as medicine or the natural sciences. Anomalous data may be useless, for example when caused by measurement errors, or it may be extremely informative and hold the key to new insights, such as very long surviving cancer patients.

Anomaly detection today has numerous applications across a great variety of domains. Examples of applications include intrusion detection in cybersecurity [418, 324, 11, 296, 593, 354], fraud detection in finance, insurance, healthcare, and telecommunication [69, 56, 260, 12, 2, 555, 617], industrial fault and damage detection [507, 438, 362, 361, 597, 343, 245, 34, 442, 615], the monitoring of infrastructure [72, 517] and stock markets [183, 184], acoustic novelty detection [437, 358, 326, 433, 287], medical diagnosis [535, 99, 318, 332, 488, 106, 246, 300, 419, 44, 489, 501, 203, 390, 554] and disease outbreak detection [582, 583], event detection in the earth sciences [64, 562, 162, 163, 587, 257], and scientific discovery in chemistry [406, 197], bioinformatics [372], genetics [546, 542], physics [89, 272], and astronomy [434, 149, 223, 453]. Detecting the *unknown unknowns* [473], often by accident, is a strong driving force in the sciences. Kuhn [293] claims that persistent anomalies drive scientific revolutions (see section VI "Anomaly and the Emergence of Scientific Discoveries" in [293]). Anomaly detection here can help us to identify new, previously unknown patterns in data, which can lead to novel scientific insights and hypotheses.

As exemplified above, anomaly detection has a broad practical impact and scientific relevance. The data available in many of these domains is continually growing in size. It is also expanding to include complex data types such as images, video, audio, text, graphs, multivariate time series, or biological sequences, among others. For applications to be successful on such complex and high-dimensional data, a meaningful representation of the data is crucial [50].

1.1.2 Why Deep Learning for Anomaly Detection?

Classic anomaly detection methods, such as the OC-SVM [496] or KDE [460, 417], often fail in high-dimensional, data-rich scenarios due to a limited computational scalability and the curse of dimensionality [241, 289, 154, 155]. To be effective, such "shallow" methods typically require substantial feature engineering.

Deep learning [307, 491, 187] follows the idea of *learning* effective representations from the data itself by training flexible, multi-layered ("deep") neural networks and has greatly improved the state of the art in many applications that involve complex data. Deep neural networks provide the most successful solutions for many tasks in

domains such as computer vision [291, 514, 532, 342, 451, 168, 218, 449, 266, 592], speech recognition [311, 121, 375, 226, 196, 211, 18, 94, 112, 492], or natural language processing [49, 370, 422, 108, 67, 261, 425, 135, 584, 79], and have contributed to the sciences [319, 40, 498, 84, 499, 262, 282, 24, 25, 157]. Methods based on deep neural networks are able to exploit the hierarchical dependencies in data because of their multi-layered, distributed feature representations. Advances in parallel computation, stochastic gradient descent optimization, and automated differentiation make it possible to apply deep learning at scale using large datasets.

Recently, there has been a surge of interest in developing deep learning approaches for anomaly detection. This interest is spurred by the lack of effective methods for anomaly detection tasks which involve complex data, for example detecting anomalous cancerous tissue from multi-gigapixel whole-slide images in histopathology [160, 58]. As in other applications of deep learning, the goal of *deep anomaly detection* 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 anomaly detection are, due to the mostly unsupervised nature of the problem.

Currently, the major approaches to deep anomaly detection include deep autoencoders [588, 91, 103, 433, 618, 623, 36, 106, 419, 1, 239, 185, 410, 394, 275], deep one-class classification [466, 476, 409, 468, 424, 423, 572, 469, 172] (the line of research to which the contents of this thesis have contributed), methods based on deep generative models such as Generative Adversarial Networks (GANs) [488, 93, 130, 13, 109, 426, 604, 489], and recent self-supervised methods [181, 222, 575, 52, 533, 521]. In comparison to traditional anomaly detection 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.

1.2 The Thesis

This thesis investigates the following hypothesis:

> *Deep One-Class Learning*, a deep learning approach to anomaly detection that is based on the one-class classification paradigm, by learning (or transferring) data representations via one-class learning objectives, can significantly improve anomaly detection performance—especially on semantic detection tasks.

Based on the results we present in this thesis, we will see that we can affirm this hypothesis. We summarize the main contributions and findings in the following.

1.2.1 Contributions and Findings

The main contributions and findings of this thesis are the following:

- We introduce Deep SVDD, one of the first deep one-class classification methods for unsupervised anomaly detection. The objective of Deep SVDD is to learn a neural network transformation that minimizes the volume of a data-enclosing hypersphere in feature space. Through this, normal data points get closely mapped to the hypersphere center, whereas anomalies are mapped away. We further identify a key challenge of deep one-class classification, namely the regularization against a trivial, constant solution, which we theoretically analyze for Deep SVDD. We demonstrate the practical value of Deep SVDD experimentally.
- We generalize Deep SVDD to the semi-supervised anomaly detection setting, where we introduce the Deep SAD method as well as Hypersphere Classification. We experimentally demonstrate the value of including two types of negative examples with these methods: (i) few labeled ground-truth anomalies, and (ii) many weakly-labeled auxiliary anomalies, which we both find can significantly improve anomaly detection performance.
- We introduce an explainable deep one-class classification variant for anomaly detection on images, called FCDD, which uses a fully convolutional architecture to incorporate the property of spatial coherence important in computer vision. For FCDD, the mapped images directly correspond to an anomaly heatmap. We evaluate the method experimentally and find that FCDD yields competitive detection performance while providing transparent explanations. In an application on detecting defects in manufacturing, FCDD achieves state-of-the-art anomaly segmentation results.
- We introduce a multi-context one-class classification variant for anomaly detection on text, called CVDD, which uses a multi-head self-attention mechanism to learn contextual sentence embeddings based on pre-trained embeddings of words. The objective of CVDD is to learn these embeddings together with a set of context vectors, such that these are closely aligned, while regularizing the context vectors to be diverse. In experiments, we find that this enables CVDD to capture multiple distinct themes present in an unlabeled text corpus, which allows to perform contextual anomaly detection.
- We present a unifying view on deep and "shallow" anomaly detection, where we distinguish the one-class classification approach from reconstruction-based methods and methods based on density estimation or generative modeling. For each of the three main approaches, we establish connections between their deep and "shallow" variants based on common underlying principles. This view contributes to a systematic understanding of existing methods and shows promising paths for future research. In a comparative evaluation, we find that the detection strategies of the various approaches are very diverse and show, using techniques for explaining anomalies, that anomaly detection models are also prone to the "Clever Hans" effect, which occurs when a model correctly detects an anomaly, but based on the "wrong" features.

Overall, the contributions and findings above demonstrate that deep one-class learning is a useful approach to anomaly detection.

1.2.2 List of Publications

The primary contributions and findings of this thesis are based on the following peer-reviewed publications:

L. Ruff*, R. A. Vandermeulen*, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep One-Class Classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4390–4399, 2018.

L. Ruff, Y. Zemlyanskiy, R. A. Vandermeulen, T. Schnake, M. Kloft. Self-Attentive, Multi-Context One-Class Classification for Unsupervised Anomaly Detection on Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4061–4071, 2019.

L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, M. Kloft. Deep Semi-Supervised Anomaly Detection. In *International Conference on Learning Representations*, 2020.

P. Liznerski*, L. Ruff*, R. A. Vandermeulen*, B. J. Franks, M. Kloft, and K.-R. Müller. Explainable Deep One-Class Classification. In *International Conference on Learning Representations*, 2021.

L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller. A Unifying Review of Deep and Shallow Anomaly Detection. *Proceedings of the IEEE*, 109(5):756–795, 2021.

The thesis also includes additional contents from the following papers:

L. Deecke, R. A. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft. Image Anomaly Detection with Generative Adversarial Networks. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 3–17, 2018.

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 *ICML 2019 Workshop on Uncertainty & Robustness in Deep Learning*, 2019.

P. Chong, L. Ruff, M. Kloft, and A. Binder. Simple and Effective Prevention of Mode Collapse in Deep One-Class Classification. In *International Joint Conference on Neural Networks*, pages 1–9, 2020.

J. R. Kauffmann, L. Ruff, G. Montavon, and K.-R. Müller. The Clever Hans Effect in Anomaly Detection. *Preprint (under review)*, 2020.

L. Ruff, R. A. Vandermeulen, B. J. Franks, K.-R. Müller, and M. Kloft. Rethinking Assumptions in Deep Anomaly Detection. In *ICML 2021 Workshop on Uncertainty & Robustness in Deep Learning*, 2021.

We note that all co-authors of these works have agreed to borrowing ideas, figures, and results from the works above for this thesis.

*Equal contribution

1.2.3 Organization of the Thesis

This thesis comprises three main chapters:

Chapter 2 (One-Class Learning) In this chapter, we introduce a deep learning approach to one-class classification. We first discuss the general one-class classification objective and briefly review established shallow one-class classification methods. We then introduce the Deep SVDD method, demonstrate theoretical properties of Deep SVDD, and evaluate the method experimentally. Afterwards, we introduce the Deep SAD method and Hypersphere Classification, which constitute generalizations of Deep SVDD to the semi-supervised setting. We present an experimental evaluation on the usefulness of having few labeled ground-truth anomalies and many weakly-labeled auxiliary anomalies available.

Chapter 3 (Applications to Computer Vision and NLP) In this chapter, we introduce two deep one-class classification variants that take advantage of their specific domains. We first introduce the FCDD method for image data, which utilizes fully convolutional networks for explainable deep one-class classification. In an experimental evaluation, we show that FCDD performs competitively while providing transparent explanations and yields state-of-the-art results on a defect detecting application in manufacturing. We then introduce the CVDD method for text data, which uses a self-attention mechanism to learn a multi-context one-class classification model. We evaluate CVDD experimentally on detecting novel topics and anomalous movie reviews.

Chapter 4 (A Unifying View of Anomaly Detection) In this chapter, we present a unifying view on deep and shallow anomaly detection methods. We first discuss methods based on density estimation and generative modeling followed by reconstruction methods, where we establish connections between their respective deep and shallow variants. We then present the unifying view, which also includes the one-class classification approach. Finally, we close this chapter with a comparative evaluation that includes canonical methods from the three main approaches (one-class classification, density estimation/generative modeling, reconstruction) which employ different feature representations (raw input, kernel, and neural network) respectively. Utilizing techniques for explaining anomalies, we demonstrate that the "Clever Hans" effect also occurs in anomaly detection.

We conclude and discuss limits of the thesis, and provide detailed paths for future research in Chapter 5. But before we turn to the main chapters of the thesis, we complete this introduction and overview with a formal introduction to the anomaly detection problem.

1.3 Formal Introduction to Anomaly Detection

In this section, we give an introduction to anomaly detection and define some terms and notation that we will use throughout the main chapters of the thesis. We first give a formal definition of the problem, then explain relevant data settings and properties as well as evaluation aspects, and finally highlight common challenges.

1.3.1 Problem Definition

In the following, 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. We then present a fundamental principle in anomaly detection, the so-called *concentration assumption*, and give a theoretical problem formulation that corresponds to density level set estimation.

What is an Anomaly?

In the introduction, we gave the following definition of an anomaly in words:

> *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 works [150, 23, 198, 216, 42], we rely on probability theory to do so.

Let $\mathcal{X} \subseteq \mathbb{R}^D$ be the space where the data lives. We define a concept of normality as the distribution $\mathbb{P}^+$ on $\mathcal{X}$ that describes 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*—can then be defined as a data point $\boldsymbol{x} \in \mathcal{X}$ (or more generally a set of points) that lies in a low probability region under $\mathbb{P}^+$. Assuming that $\mathbb{P}^+$ has a corresponding probability density function (pdf) $p^+(\boldsymbol{x})$, we define the *set of anomalies* as

$$\mathcal{A} = \{\boldsymbol{x} \in \mathcal{X} \mid p^+(\boldsymbol{x}) \leq \tau\}, \quad \tau \geq 0, \tag{1.1}$$

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

Types of Anomalies

Various types of anomalies have been identified in the literature [95, 6]. These include point anomalies, conditional or contextual anomalies [523, 520, 96, 205, 14, 344, 480], and group or collective anomalies [96, 594, 382, 601, 70, 93]. In [470], we have extended these three established types with low-level sensory anomalies and high-level semantic anomalies [9], a distinction that is particularly relevant for deciding between deep and shallow methods.

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 anomaly detection research.

A *conditional* or *contextual anomaly* is a data instance that is anomalous within 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 type of anomaly, the normal law $\mathbb{P}^{\blacktriangle}$ is more precisely a conditional distribution $\mathbb{P}^{\blacktriangle} \equiv \mathbb{P}^{\blacktriangle}_{X|T}$ with conditional pdf $p^{\blacktriangle}(\boldsymbol{x} \,|\, t)$ that depends on some contextual variable T. Time-series anomalies [165, 550, 551, 205, 302, 480] are the most prominent example of contextual anomalies. Other examples include spatial [100, 497], spatio-temporal [520], or graph-based [397, 14, 235] 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 is anomalous, where $J \subseteq \mathbb{N}$ is an index set that captures some relation or dependency. A cluster of anomalies such as similar or related network attacks in cybersecurity form a collective anomaly for instance [296, 235, 10]. Often, collective anomalies are also contextual such as anomalous time series or biological sequences, for example a series or sequence $\{\boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t+s-1}\}$ of length $s \in \mathbb{N}$. Here, 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 (stationary) marginal $p^{\blacktriangle}(\boldsymbol{x}) = \int p^{\blacktriangle}(\boldsymbol{x}, t)\,\mathrm{d}t$ or under the sequence-integrated, time-conditional marginal $p^{\blacktriangle}(\boldsymbol{x}_j \,|\, t)$ given by

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

the full series or sequence $\{\boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t+s-1}\}$ can be anomalous under the *joint* time-conditional density $p^{\blacktriangle}(\boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t+s-1} \,|\, t)$, which appropriately describes the (conditional) 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* [9] 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 texts. 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) [50, 340]. We can express this dependence via a law of normality with conditional pdf $p^{\blacktriangle}(\boldsymbol{x} \,|\, \boldsymbol{z}, y)$, where we usually assume Z to be continuous and Y to be discrete. Texture defects and pixel artifacts in images, or character typos in words, are both examples of low-level anomalies. In comparison, images of non-normal objects [9] for instance, or misposted reviews and news articles [468], are examples of semantic anomalies. Note that a semantic anomaly may be very close to normal instances in the raw feature

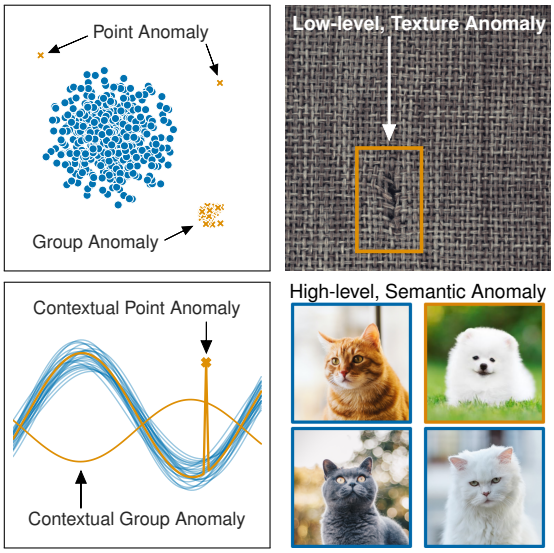


Figure 1.1: An illustration of the various types of anomalies: A *point anomaly* is a single anomalous point. A *contextual point anomaly* occurs if a point deviates in its local context, here a spike in an otherwise normal time series. A *group anomaly* can be a cluster of anomalies or some series of related points that is 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 in the low-level features, here a cut in the fabric texture of a carpet [54]. 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.

space $\mathcal{X}$. An image of a dog with a fur color and texture similar to that of some cat, for example, can be more similar in raw pixel space than various cat breeds among themselves (see Figure 1.1). Likewise, low-level background statistics can result in a high similarity in raw pixel space even when objects in the foreground are completely different [9]. 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, triangular snout, etc.), which is an inherently difficult task in an unsupervised setting [340]. On the other hand, sensory anomalies may also be very close to normal instances in some semantic feature space $\mathcal{Z}$. An image of a carpet with a small cut in the fabric may semantically still be closely identified as "carpet." An awareness of this type distinction (sensory vs. semantic) is therefore critical for successful applications.

Anomaly, Outlier, or Novelty?

Some works 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 $\mathcal{A}$ as defined in (1.1)), an anomaly is often characterized as being an instance from a distribution that is truly different from

$\mathbb{P}^+$ (e.g., when anomalies are generated by a different process than normal data), 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, non-stationary $\mathbb{P}^+$. 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 (see Figure 1.2). Such a distinction between anomaly, outlier, and novelty may reflect slightly different objectives in an application. Whereas anomalies are often the data points of interest (e.g., a long-term survivor of a deadly disease), outliers are frequently regarded as "noise" or "measurement error" that should be removed in a data pre-processing 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 for the most part the same, however. For this reason, we make no distinction between these terms in this thesis and call any instance $\boldsymbol{x} \in \mathcal{A}$ an "anomaly."

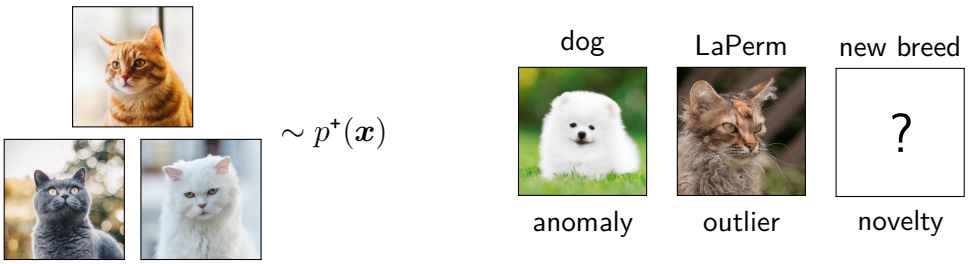


Figure 1.2: An example for the difference between an "anomaly," an "outlier," and a "novelty." Under the normal law p^+ of cats, the dog would be an anomaly (not being a cat), the LaPerm would be an outlier (being a rare cat breed), and a new breed of cats would be a novelty.

The Concentration Assumption

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

$$\mathcal{X} \setminus \mathcal{A} = \{\boldsymbol{x} \in \mathcal{X} \mid p^+(\boldsymbol{x}) > \tau\} \tag{1.2}$$

is non-empty and small (typically in the Lebesgue-measure sense, i.e. the ordinary notion of volume in D-dimensional space). This is also known as the so-called *concentration* or *cluster assumption* [493, 525, 97]. Note that the concentration assumption does not imply that the full support $\operatorname{supp}(p^+) = \{\boldsymbol{x} \in \mathcal{X} \mid p^+(\boldsymbol{x}) > 0\}$ of the normal law $\mathbb{P}^+$ must be bounded; only that some high-density subset of the support can be bounded. The support of a standard univariate Gaussian 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 $\mathcal{A}$ must not be concentrated and can be unbounded.

Density Level Set Estimation

A law of normality $\mathbb{P}^+$ is only known in rare cases, such as for certain laws of physics. Sometimes a concept of normality can 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 usually 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 further below.

Given data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ generated by $\mathbb{P}$ (usually assumed to be drawn from i.i.d. random variables following $\mathbb{P}$), the goal of anomaly detection 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 anomaly detection 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* [431, 553, 47, 457] which corresponds to *minimum volume set estimation* [432, 500] for the special case in which sets are defined via probability density values. The density level set of $\mathbb{P}$ for some threshold $\tau \geq 0$ is given by $C = \{\boldsymbol{x} \in \mathcal{X} \mid p(\boldsymbol{x}) > \tau\}$. 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,

$$\begin{aligned} C_\alpha &= \operatorname*{arginf}_C \{\lambda(C) \mid \mathbb{P}(C) \geq 1 - \alpha\} \\ &= \{\boldsymbol{x} \in \mathcal{X} \mid p(\boldsymbol{x}) > \tau_\alpha\} \end{aligned} \tag{1.3}$$

where $\tau_\alpha \geq 0$ denotes the corresponding threshold and $\lambda(\cdot)$ is typically again the Lebesgue measure. The extreme cases of $\alpha = 0$ and $\alpha \to 1$ result in the full support $C_0 = \{\boldsymbol{x} \in \mathcal{X} \mid p(\boldsymbol{x}) > 0\} = \operatorname{supp}(p)$ 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 and threshold τ_α such that a corresponding level set C_α exists and can be bounded. Figure 1.3 illustrates some density level sets for the familiar case where $\mathbb{P}$ is the standard Gaussian distribution. Given a level set C_α, we can define the corresponding threshold anomaly detector $c_\alpha : \mathcal{X} \to \{\pm 1\}$ as

$$c_\alpha(\boldsymbol{x}) = \begin{cases} +1 & \text{if } \boldsymbol{x} \in C_\alpha, \\ -1 & \text{if } \boldsymbol{x} \notin C_\alpha. \end{cases} \tag{1.4}$$

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

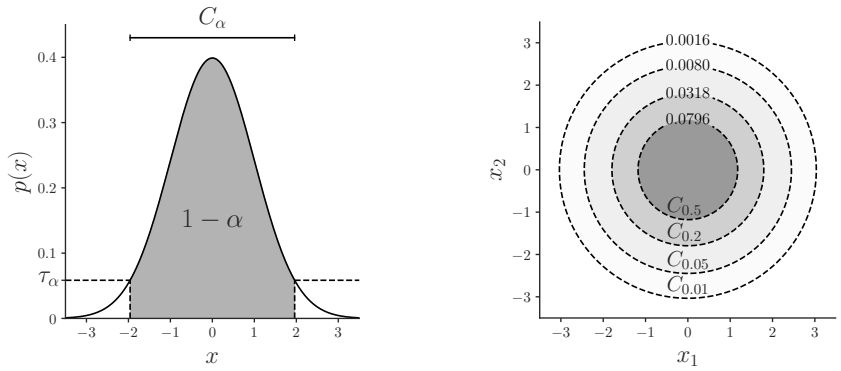


Figure 1.3: An illustration of the α-density level sets C_α with threshold τ_α for a univariate (left) and bivariate (right) standard Gaussian distribution.

level $\alpha \in [0, 1]$, one can estimate a corresponding threshold $\hat{\tau}_\alpha$ via the empirical p-value:

$$\hat{\tau}_\alpha = \inf_\tau \left\{ \tau \geq 0 \;\middle|\; \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{[0,\hat{p}(\boldsymbol{x}_i))}(\tau) \geq 1 - \alpha \right\}, \tag{1.5}$$

where $\mathbb{1}_A(\cdot)$ denotes the indicator function for some set A. Using $\hat{\tau}_\alpha$ and $\hat{p}(\boldsymbol{x})$ in (1.3) yields the plug-in density level set estimator $\hat{C}_\alpha$ which can be used in (1.4) to obtain the plug-in threshold detector $\hat{c}_\alpha(\boldsymbol{x})$. Note that density estimation is generally the most costly approach to density level set estimation (in terms of required data samples), since estimating the full density is equivalent to first estimating the *entire family* of level sets $\{C_\alpha \mid \alpha \in [0, 1]\}$ from which the desired level set for some fixed $\alpha \in [0, 1]$ is then only selected [173, 367]. If there are insufficient samples, this density estimate can be biased. This is one of the reasons that motivated the development of one-class classification methods which aim to estimate a collection [367] or single level sets [537, 536, 496, 538] directly, which we cover in Chapter 2 in detail.

Threshold vs. Score

Approaching density level set estimation through density estimation is relatively costly, as mentioned above, yet this approach results in a more informative model that allows to rank inliers and anomalies (inside and outside the level set) according to the estimated normal data density. In comparison, a pure threshold detector as in (1.4) only yields a binary prediction. Menon and Williamson [367] have proposed a compromise by learning a density outside the level set boundary. Many anomaly detection 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 [115, 180]. 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 anomaly detection applications. As for an estimated density, of course, we can always derive a threshold as in (1.5) from the empirical distribution of anomaly scores if needed.

Selecting a Level α

There are many degrees of freedom when tackling a specific anomaly detection 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). Choosing α also involves further assumptions about the data-generating process $\mathbb{P}$, which we have assumed to match the normal data distribution $\mathbb{P}^+$ so far. In the next section, we discuss data settings and properties in anomaly detection that may alter this assumption.

1.3.2 Data Settings and Properties

The dataset settings (e.g., unsupervised or semi-supervised) and intrinsic data properties (e.g., dimensionality or feature type) that occur in anomaly detection applications can be diverse. We here characterize these settings which can range from the standard unsupervised to a semi-supervised as well as a supervised setting and list further data properties that are relevant for modeling a specific anomaly detection problem. But first we observe that the assumptions made about the distribution of anomalies (often implicitly) are also crucial to the problem.

A Distribution of Anomalies?

Let $\mathbb{P}^-$ denote the *ground-truth anomaly distribution* and assume that it exists on data space $\mathcal{X} \subseteq \mathbb{R}^D$. 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 [493, 525]. This assumption may be modeled by an anomaly distribution $\mathbb{P}^-$ that follows a uniform distribution over the (bounded[1]) data space $\mathcal{X}$ [536]. Some well-known unsupervised methods such as KDE [417] or the OC-SVM [496], for example, can be interpreted as implicitly making this assumption that $\mathbb{P}^-$ follows a uniform distribution which can be viewed as a default uninformative prior on the anomalous distribution [525]. This standard prior assumes that there are no anomalous modes and that anomalies are equally likely to occur over the data space $\mathcal{X}$. Semi-supervised or supervised anomaly detection

[1]Strictly speaking, we here assume 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.

approaches often depart from this uninformed prior and try to make a more informed a-priori assumption about the anomalous distribution $\mathbb{P}^-$ [525]. If truthful to $\mathbb{P}^-$, such a model based on a more informed anomaly prior can achieve better detection performance. Modeling anomalous modes also can be useful in certain applications, for example, for learning typical modes of failure in industrial machines or known disorders in medical diagnosis. Note that such prior assumptions about the anomaly distribution $\mathbb{P}^-$ are often expressed only implicitly in the literature, though these assumptions are critical to anomaly detection modeling.

The Unsupervised Setting

The unsupervised anomaly detection setting is the case in which only unlabeled data

$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \tag{1.6}$$

is available for training a model. This setting is arguably the most common setting in anomaly detection [232, 95, 621, 427]. Typically, we assume that these data points have been drawn in an i.i.d. fashion from the data-generating distribution $\mathbb{P}$. For simplicity, we so far have assumed that the data-generating distribution is the same as the normal data distribution $\mathbb{P} \equiv \mathbb{P}^+$. This is often expressed with the statement that the training data is assumed to be "clean." In practice, however, the data-generating distribution $\mathbb{P}$ may be subject to 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 $\mathbb{P}$ are of the form $\boldsymbol{x} + \varepsilon$ where $\boldsymbol{x} \sim \mathbb{P}^+$. Noise might be present due to irreducible measurement uncertainties in an application, for instance. The greater the noise, the harder it becomes to accurately estimate the ground-truth level sets of $\mathbb{P}^+$, since informative normal features get obfuscated [621]. 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 disturbance. For example, some anomalous degradation in an industrial machine might already occur unnoticed 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 $\mathbb{P} \equiv (1-\gamma_p)\,\mathbb{P}^+ + \gamma_p\,\mathbb{P}^-$ with contamination (or pollution) rate $\gamma_p \in (0, 1)$. The greater the contamination, the more a normal data decision boundary may get distorted by the (undetected) anomalous points when learning a model.

In summary, a more general and realistic assumption for the data-generating distribution $\mathbb{P}$ is that data samples have the form of $\boldsymbol{x} + \varepsilon$ where $\boldsymbol{x} \sim (1 - \gamma_p)\,\mathbb{P}^+ + \gamma_p\,\mathbb{P}^-$ and ε is random noise. Both, assumptions on the noise distribution ε and contamination rate γ_p, are therefore important for modeling a specific anomaly detection problem. Robust methods [210, 244, 618] specifically aim to account for these sources of disturbance. Also note 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.

The Semi-Supervised Setting

The semi-supervised anomaly detection setting is the case in which both unlabeled and labeled data

$$\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X} \quad \text{and} \quad (\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \dots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y} \tag{1.7}$$

are available for training a model with $\mathcal{Y} = \{\pm 1\}$, where $\tilde{y} = +1$ denotes normal and $\tilde{y} = -1$ denotes anomalous points respectively.

Usually, we have $m \ll n$ in the semi-supervised setting, that is, most of the data is unlabeled and only a few labeled instances are available, since labels are often costly to obtain in terms of resources (time, money, etc.). Labeling, for instance, may require domain experts such as medical professionals (e.g., pathologists) or technical experts (e.g., aerospace engineers). Anomalies, in particular, are also rare by nature (e.g., rare medical conditions) or very costly (e.g., the failure of some industrial machine). Deliberately generating anomalies is therefore usually not a practical option. However, including some known anomalous examples, if available, can already significantly improve the detection performance of a model (see Section 2.3 and [536, 338, 191, 371, 280, 469]). Labels might be available in monitoring tasks, for example, where alarms raised by an anomaly detector have been examined to see whether they were correct. Some unsupervised anomaly detection methods can be incrementally updated when such labels become available [512].

Another recent idea, called *Outlier Exposure* (OE) [221] uses large quantities of unlabeled data that is available in some domains as auxiliary anomalies (e.g., online stock photos in computer vision or the English Wikipedia in NLP), thereby effectively labeling this data with $\tilde{y} = -1$. In this setting, we frequently have that $m \gg n$, but this labeled data has a higher labeling uncertainty as the auxiliary data may not only contain anomalies and may not be representative of anomalies at testing time. Along with introducing novel methods that can particularly incorporate labeled data into learning, we will assess the usefulness of few labeled true anomalies (Section 2.3.4 and Section 3.1.2) and many weakly-labeled auxiliary anomalies (Section 2.3.6 and Section 3.1.2) in various experiments covering different scenarios in this thesis.

Verifying unlabeled samples as indeed being normal is often more viable due to the more frequent nature of normal data. This is one reason why the special semi-supervised case of *Learning from Positive and Unlabeled Examples* (LPUE) [134, 607, 145] (i.e., labeled normal and unlabeled examples) is also specifically studied in the anomaly detection literature [95, 384, 63, 522, 13]. Previous work [95] has also referred to the special case of learning exclusively from positive examples as the "semi-supervised anomaly detection" setting. Although meticulously curated normal data can sometimes be available (e.g., in open category detection [335]), such a setting in which entirely (and confidently) labeled normal examples are available is rather rare in practice. The analysis of this setting is again rather justified by

the *assumption* that most of the given (unlabeled) training data is 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 [95]. We therefore refer to the more general setting as presented in (1.7) as the semi-supervised anomaly detection setting, which incorporates both labeled normal and anomalous examples in addition to unlabeled data points, 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 above, 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).

The Supervised Setting

The supervised anomaly detection setting is the case in which completely labeled data

$$(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \dots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y} \tag{1.8}$$

is 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 anomaly detection problem, but a classification task. Although anomalous modes or clusters may exist in an application, that is, some anomalies may 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) classes, but in anomaly detection the objective remains the estimation of the normal data density level set boundaries. Hence, we should interpret the supervised anomaly detection problem as "label-informed density level set estimation" in which normal (in-distribution) and anomalous (out-of-distribution) training examples are available. Due to the above, and also the high costs that are often involved with labeling, the supervised anomaly detection setting is the most uncommon setting in practice.

We finally 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 out-of-distribution detection [219] or open category detection [335] problems, for example, the goal is to train a classifier while also detecting examples that are not from any of the known set of training classes. In these problems, the labeled data $(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \dots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m)$ with $\tilde{y} \in \{1, \dots, k\}$ contains additional information about some k (sub-)classes of the in-distribution $\mathbb{P}^+$. Including such information about the structure of $\mathbb{P}^+$ has been shown to be beneficial for semantic detection tasks

[101, 487]. We will discuss this connection of detection problems also in our outlook, as an opportunity to bridging related lines of research (see Section 5.2.2).

Intrinsic Data Properties

Besides the dataset settings described above, the intrinsic properties of the data itself are also crucial for modeling a specific anomaly detection problem. Table 1.1 provides a list of data properties that can be relevant to consider. In Figure 1.4, we further show a toy dataset with a specific realization of these properties, which will serve us as a running example in this thesis. The assumptions about the intrinsic data properties should be reflected in the modeling choices, such as adding context or choosing between deep or shallow feature maps, which can be challenging. We outline these and further challenges in anomaly detection in Section 1.3.4, after having discussed the evaluation of anomaly detection methods in the next Section 1.3.3.

Table 1.1: Data properties relevant to consider in anomaly detection.

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., as in 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? At what contamination rate γ_p?

1.3.3 Evaluation

There are two types of errors an anomaly detection model can make: (i) predicting a true normal data point as being anomalous (*type I error* or *false alarm*), and (ii) predicting a true anomaly as being normal (*type II error* or *missed anomaly*). There is no universal "golden rule" to balance these two types of errors, since the costs that are associated with each type can vary depending on the application. In medical

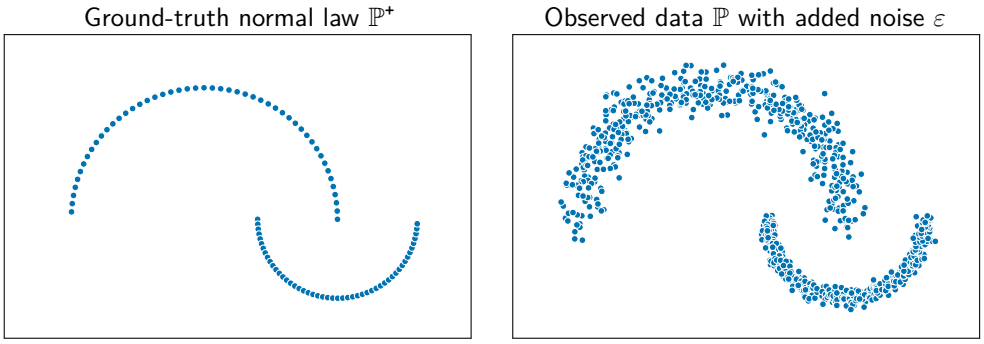


Figure 1.4: A two-dimensional *Big Moon, Small Moon* toy example with real-valued ground-truth normal law $\mathbb{P}^+$ that is composed of two one-dimensional manifolds (bimodal, two-scale, non-convex). The unlabeled training data ($n = 1{,}000$, $m = 0$) is generated from $\mathbb{P}$ that adds Gaussian noise ε to normal samples from $\mathbb{P}^+$. This toy data is non-hierarchical, context-free, and stationary. Anomalies are off-manifold points that may occur uniformly over the displayed range.

diagnosis, for example, the costs of missing an anomaly (e.g., missing cancerous tissue) are often greater, which is why a low miss rate, usually at the expense of a higher false alarm rate, is often desirable. In monitoring tasks that involve large amounts of data, on the other hand, it can be more desirable to have a low false alarm rate (e.g., to present anomalies with high accuracy to analysts for opening an investigation), usually at the expense of a higher miss rate. Given the anomaly score $s : \mathcal{X} \to \mathbb{R}$ of some model, the decision threshold τ with

$$\text{decide} \begin{cases} \text{anomaly} & \text{if} \quad s(\boldsymbol{x}) \geq \tau, \\ \text{inlier} & \text{if} \quad s(\boldsymbol{x}) < \tau, \end{cases} \tag{1.9}$$

should therefore be calibrated for the specific application, so that the costs associated with type I and type II errors are minimized (e.g., using some labeled validation data), or possible hard constraints of the application are met.

For a systematic evaluation of anomaly detection models, or when application-related costs and constraints are not fully specified, it is more desirable to consider a measure that evaluates the performance over a broad range of possible application scenarios, or analogously, a broad range of decision thresholds τ. The Area Under the Receiver Operating Characteristic curve (AUROC), usually simply called Area Under the Curve (AUC), provides such an evaluation measure that considers the full range of decision thresholds on a given test set [76]. The Receiver Operating Characteristic (ROC) curve [161] plots all (false alarm rate, recall)-pairs that result from iterating over all possible thresholds covering every 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 (or 50%), regardless of whether there is an imbalance between anomalies and normal instances in the test set. This makes the AUC easy to interpret and comparable over different application scenarios, which is one of the reasons why the AUC is the standard performance measure used in anomaly detection [139, 81].

One caveat of the AUC is that it can produce overly optimistic scores in the case of highly imbalanced test sets [129, 9]. In such cases, the Area Under the Precision-Recall Curve (AUPRC) is more informative and appropriate to use [129, 9]. The Precision-Recall (PR) curve plots all the (precision, recall)-pairs that result from iterating over all possible test set decision thresholds. The AUPRC therefore is preferable to the AUROC when precision is more relevant than the false alarm rate. A common robust way to compute the AUPRC is via Average Precision (AP) [75]. One downside of the 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 the AUPRC (or AP) generally harder to interpret and less comparable over different application scenarios.

In cases where the test set is not highly imbalanced, the AUROC and AUPRC (or AP) measures show the same trends [129]. To exemplify this practically, we report the results for both measures for the comparative evaluation in Section 4.4. Since the test sets of the datasets and benchmarks presented in this thesis are overall not highly imbalanced, however, we mainly report the results of our experiments using the standard AUROC measure, denoted as AUC.

1.3.4 Challenges

We conclude this introduction by highlighting some notable challenges in anomaly detection, some of which directly arise from the definition and data characteristics given above. Certainly, the fundamental challenge in anomaly detection 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, potential prior assumptions about the anomaly distribution, and, if available, the challenge of how to incorporate labeled data instances in a generalizing manner. Further questions include how an anomaly score or threshold can be derived in a specific task? What level α 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, that is, is robustness a critical aspect? Moreover, identifying and including the data properties given in Table 1.1 into a method and model can pose challenges as well. The computational complexity in both the dataset size $n+m$ and dimensionality D as well as the memory cost of a model at training time, but also at testing time can be a limiting factor (e.g., for data streams in real-time monitoring [71]). Is the data-generating process assumed to be non-stationary [529, 436, 528] or are there distributional shifts expected at testing time? For (truly) high-dimensional data, the curse of dimensionality and the resulting concentration of distances can be a major issue [621]. Finding a meaningful data representation that captures the relevant features that are useful for a given detection task and domain here becomes vital. Deep anomaly detection methods further entail new challenges such as an increased number of hyperparameters and the selection of suitable network architectures and optimization parameters (learning rate, batch sizes, etc.). In addition, the more complex the data or a model is, the

greater the challenges of model interpretability (e.g., [38, 377, 299, 481]) and decision transparency become.

Considering the various facets of the anomaly detection problem we have covered in this introduction, it is perhaps not surprising that such a wealth of literature and variety of methods for this problem exist. We now turn to the three main chapters and contributions of this thesis to this vivid field of research. In the following Chapter 2, we will dive into one particular approach to anomaly detection, namely one-class classification. The main contribution of this chapter lies in the extension of the one-class classification approach from fixed features towards learning representations (utilizing deep neural networks) that prove useful for the anomaly detection task. Prior to the introduction of "Deep One-Class Classification" [466], finding useful features was mostly treated as a separate pre-processing step in the anomaly detection literature ("feature engineering"), after which one-class classification models were then subsequently trained using the pre-selected (fixed) features. In Chapter 3, we will then consider specific applications of deep one-class learning to computer vision and NLP. The contribution of this chapter lies in the introduction of two domain-specific methods, *Fully Convolutional Data Description* for images and *Context Vector Data Description* for text, both of which, by construction, allow for model interpretability. In Chapter 4, we will first examine two other main approaches to anomaly detection, probabilistic methods (density estimation and generative modeling) and reconstruction methods. Afterwards, we will present a unifying view of anomaly detection methods, specifically identifying connections between deep and shallow methods, which will enable us to systematically identify open challenges and paths for future research.

2 One-Class Learning

In this chapter, we introduce a deep learning approach to one-class classification. We term this approach of learning representations (e.g., utilizing deep neural networks) for one-class classification *One-Class Learning*. We first briefly review established shallow one-class classification methods in Section 2.1. In Section 2.2, we then introduce the Deep SVDD method. We demonstrate theoretical properties of Deep SVDD and evaluate the method experimentally. Here, we also identify a key challenge in deep one-class classification, which is the regularization against a trivial, constant solution. In Section 2.3, we turn to the semi-supervised anomaly detection setting and introduce the Deep SAD method (Section 2.3.2) as well as Hypersphere Classification (Section 2.3.5), which both constitute semi-supervised generalizations of Deep SVDD. In an experimental evaluation of these semi-supervised methods, we find that even few labeled true anomalies (Section 2.3.4) or many weakly-labeled auxiliary anomalies (Section 2.3.6) can significantly improve anomaly detection performance.

2.1 Shallow One-Class Classification

One-class classification [379, 378, 537, 536, 271], occasionally also called *single-class classification* [373, 151], adopts a discriminative approach to anomaly detection. Methods based on one-class classification try to avoid a full estimation of the probability density as an intermediate step to anomaly detection. Instead, these methods aim to directly learn a decision boundary that corresponds to some desired density level set of the normal data distribution $\mathbb{P}^+$.

2.1.1 One-Class vs. Binary Classification

We can view 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 (i) falsely raised alarms for true normal instances (i.e., the false alarm rate or type I error), and (ii) 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

data points, for example a sufficiently large ball that contains all points. However, the crux here of course is to simultaneously keep the miss rate low, that is, to not draw this boundary too loosely. For this reason, one often specifies some target false alarm rate $\alpha \in [0, 1]$ *a priori* in an application, for which the miss rate is then sought to be minimized. Note that this exactly corresponds to the idea of estimating an α-density level set for some a priori fixed level $\alpha \in [0, 1]$, as described in the introduction (see Section 1.3). 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 few) anomalies.

We can express the rationale above in terms of the binary classification risk [525, 367]. 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 (e.g., hinge, logistic, etc.) 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:

$$R(f) = \mathbb{E}_{X \sim \mathbb{P}^{+}}[\ell(f(X), +1)] + \mathbb{E}_{X \sim \mathbb{P}^{-}}[\ell(f(X), -1)]. \tag{2.1}$$

Minimizing the first term, the expected loss of classifying true normal instances as anomalous, corresponds to minimizing the (expected) false alarm rate. Minimizing the second term, the expected loss of classifying true anomalies as normal, corresponds to minimizing the (expected) miss rate.

Given some unlabeled data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$, and potentially some additional labeled data $(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \ldots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m)$, we can apply the principle of empirical risk minimization to obtain

$$\min_{f} \quad \frac{1}{n} \sum_{i=1}^{n} \ell(f(\boldsymbol{x}_i), +1) + \frac{1}{m} \sum_{j=1}^{m} \ell(f(\tilde{\boldsymbol{x}}_j), \tilde{y}_j) + \mathcal{R}. \tag{2.2}$$

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 (2.2) would be trivial in this setting. We add $\mathcal{R}$ as an additional term to denote and capture such regularization which can take various forms depending on the assumptions about f, but critically also about $\mathbb{P}^{-}$. Generally, the regularization $\mathcal{R} = \mathcal{R}(f)$ aims at minimizing the miss rate (e.g., via volume minimization and assumptions about $\mathbb{P}^{-}$) and improve generalization (e.g., via smoothing of f). Further note, that the pseudo-labeling of $y = +1$ in the first term incorporates the assumption that the n unlabeled training data points are normal (i.e., have been drawn from $\mathbb{P}^{+}$). This assumption can be altered through specific choices of the loss and regularization. For example, by requiring some fraction of the unlabeled data to get misclassified to incorporate some assumption about the contamination rate γ_p or to achieve some target false alarm rate α.

In summary, in contrast to standard binary classification, where we have data from both classes to learn a discriminative decision function f that minimizes the

risk $R(f)$ and we typically can assume both classes to have some distinct structure (e.g., being concentrated in data space $\mathcal{X}$), we generally lack such information in one-class classification. Instead, we commonly have data from only one class (with increased uncertainty, if the data is unlabeled) and little to no knowledge about the anomalies, which necessarily requires making certain modeling assumptions.

2.1.2 One-Class Classification in Input Space

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

$$\begin{aligned} \min_{R, \boldsymbol{c}, \boldsymbol{\xi}} \quad & R^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & \|\boldsymbol{x}_i - \boldsymbol{c}\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \quad \forall i \in \{1, \dots, n\}. \end{aligned} \tag{2.3}$$

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

Objective (2.3) exactly corresponds to SVDD applied in the input space $\mathcal{X}$, motivated above as in [537, 536, 538]. Equivalently, we can derive (2.3) also from the classification risk (2.1). Consider the (shifted, cost-weighted) hinge loss $\ell(s, y)$ [367] defined by:

$$\ell(s, y) = \begin{cases} \frac{1}{1+\nu} \max(0, s) & \text{if } y = +1, \\ \frac{\nu}{1+\nu} \max(0, -s) & \text{if } y = -1. \end{cases} \tag{2.4}$$

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 minimization objective (2.1) (multiplied by $1+\nu$) is given by:

$$\min_{\theta} \quad \mathbb{E}_{X \sim \mathbb{P}^+}[\max(0, \|X - \boldsymbol{c}\|^2 - R^2)] + \nu \, \mathbb{E}_{X \sim \mathbb{P}^-}[\max(0, R^2 - \|X - \boldsymbol{c}\|^2)]. \tag{2.5}$$

We can estimate the first term of (2.5) empirically from $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$, again assuming (most of) these points have been drawn from $\mathbb{P}^+$. If labeled anomalies are absent, we still can 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:

$$\begin{aligned} \mathbb{E}_{X \sim \mathcal{U}(\mathcal{X})}[\max(0, R^2 - \|X - \boldsymbol{c}\|^2)] &= \frac{1}{\lambda(\mathcal{X})} \int_{\mathcal{X}} \max(0, R^2 - \|\boldsymbol{x} - \boldsymbol{c}\|^2) \, \mathrm{d}\lambda(\boldsymbol{x}) \\ &\leq R^2 \, \frac{\lambda(\mathcal{B}_R(\boldsymbol{c}))}{\lambda(\mathcal{X})} \leq R^2, \end{aligned} \tag{2.6}$$

where $\mathcal{B}_R(\boldsymbol{c}) \subseteq \mathcal{X}$ denotes the ball centered at $\boldsymbol{c}$ with radius R and $\lambda(\cdot)$ is again the standard (Lebesgue) measure of volume.[1] This shows that the minimum volume principle [432, 500] can be naturally motivated in one-class classification by seeking to minimize the risk of missing anomalies, here illustrated for the assumption that the anomalies follow a uniform distribution. Using the simple upper bound (2.6) for the second term in (2.5), we thus can derive the following overall empirical objective:

$$\min_{R,\boldsymbol{c}} \quad R^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \max(0, \|\boldsymbol{x}_i - \boldsymbol{c}\|^2 - R^2). \tag{2.7}$$

This is equivalent to (2.3), but with the constraints directly incorporated into the objective function. We remark that the cost-weighting hyperparameter $\nu \in (0, 1]$ is purposefully chosen here, since it is an upper bound on the ratio of points outside and a lower bound on the ratio of points that are outside or on the boundary of the sphere, the so-called ν-property (see Proposition 4 or [496] for a proof). 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 a very limited model, of course, and only matches a limited class of distributions $\mathbb{P}^+$ (e.g., an isotropic Gaussian distribution). Minimum Volume Ellipsoids (MVE) [463, 465] and the Minimum Covariance Determinant (MCD) estimator [464] are a generalization to non-isotropic distributions with elliptical support. Nonparametric methods such as One-Class Neighbor Machines [381] provide additional freedom to model multimodal distributions having non-convex support. Extending the objective and principles above to general feature spaces (e.g., [560, 495, 493]) further increases the flexibility of one-class models and enables to learn decision boundaries for more complex distributions.

2.1.3 Kernel-Based One-Class Classification

The kernel-based OC-SVM [496] and SVDD [536, 538] are perhaps the most well-known one-class classification methods. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be some positive semi-definite (PSD) kernel with associated reproducing kernel Hilbert space (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 $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{X}$ [27]. The objective of (kernel) SVDD is again to find a data-enclosing hypersphere of minimum volume. The SVDD primal problem is the one formulated in (2.3), but with using a hypersphere model $f_\theta(\boldsymbol{x}) = \|\phi_k(\boldsymbol{x}) - \boldsymbol{c}\|^2 - R^2$ defined in kernel 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:

$$\begin{aligned} \min_{\boldsymbol{w},\rho,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 - \rho + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & \rho - \langle \phi_k(\boldsymbol{x}_i), \boldsymbol{w} \rangle \leq \xi_i, \quad \xi_i \geq 0, \quad \forall i \in \{1, \ldots, n\}. \end{aligned} \tag{2.8}$$

[1] Note again that we here assume that the data space $\mathcal{X}$ can be bounded to numerically meaningful values, so that $\lambda(\mathcal{X}) < \infty$ and thus the uniform distribution $\mathcal{U}(\mathcal{X})$ is well-defined.

So 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 $\frac{\rho}{\|\boldsymbol{w}\|}$ which is maximized through maximizing ρ, where $\|\boldsymbol{w}\|$ acts as a normalizer.

The OC-SVM and SVDD both can be solved in their respective dual formulations, which are quadratic programs that only involve dot products (the feature map ϕ_k is implicit), for instance by applying sequential minimal optimization [428]. For the standard Gaussian/RBF kernel (or any kernel with constant norm $k(\boldsymbol{x}, \boldsymbol{x}) = c > 0$), the OC-SVM and SVDD are equivalent [536]. In this case, the corresponding density level set estimator defined by

$$\hat{C}_\nu = \{\boldsymbol{x} \in \mathcal{X} \mid f_{\hat{\theta}}(\boldsymbol{x}) < 0\}, \tag{2.9}$$

with estimated model parameters $\hat{\theta}$, is in fact an asymptotically consistent ν-density level set estimator [564]. The solution paths of hyperparameter ν have been analyzed for both the OC-SVM [309] and SVDD [518].

Kernel-induced feature spaces considerably improve the expressive power of one-class methods and allow to learn well-performing models in multimodal, non-convex, and non-linear 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 [308, 176], Multi-Sphere SVDD [192], Multiple Kernel Learning for OC-SVM [126, 169], OC-SVM for group anomaly detection [382], boosting via L^1-norm regularized OC-SVM [446], One-Class Kernel Fisher Discriminants [461, 462, 148], Bayesian Data Description [174], as well as distributed [527], incremental learning [256], or robust [336] variants.

2.2 Deep One-Class Classification

Selecting kernels and hand-crafting relevant features can be challenging and quickly become impractical for complex data. With deep one-class classification [466], we aim to overcome these challenges by learning useful representations via deep neural networks from the data or transferring such representations from related tasks.

2.2.1 The Deep SVDD Method

We here introduce *Deep Support Vector Data Description* (Deep SVDD), a method for deep one-class classification that is inspired by kernel-based one-class classification and minimum volume estimation [432, 500]. Deep SVDD trains a neural network $\phi_\omega : \mathcal{X} \to \mathcal{Z}$ with network weights ω while minimizing the volume of a data-enclosing hypersphere in output space $\mathcal{Z}$ (see Figure 2.1). Minimizing the volume of the hypersphere forces the network to extract common factors of variation of the data since the network must closely map the data points to the center of the sphere.

For input space $\mathcal{X} \subseteq \mathbb{R}^D$ and output space $\mathcal{Z} \subseteq \mathbb{R}^d$, let $\phi_\omega : \mathcal{X} \to \mathcal{Z}$ be a neural network with $L \in \mathbb{N}$ layers and set of weights $\omega = \{\omega_1, \ldots, \omega_L\}$ where ω_l are the weights of layer $l \in \{1, \ldots, L\}$. That is, $\phi_\omega(\boldsymbol{x}) \in \mathcal{Z}$ is the feature representation of

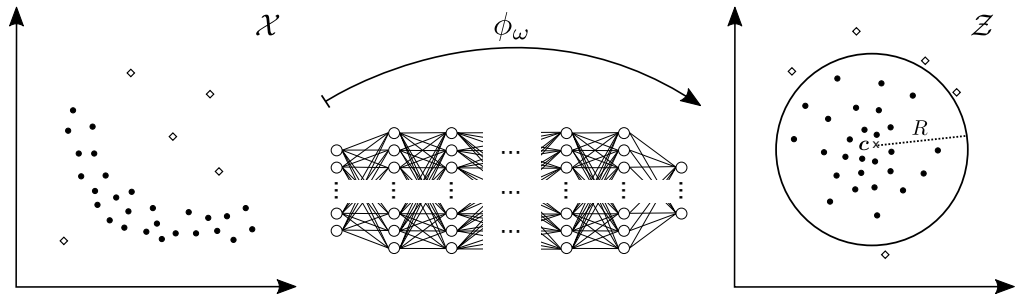


Figure 2.1: Deep SVDD learns a neural network transformation ϕ_ω with weights ω from input space $\mathcal{X} \subseteq \mathbb{R}^D$ to output space $\mathcal{Z} \subseteq \mathbb{R}^d$ that attempts to map the data points into a hypersphere characterized by center $\boldsymbol{c}$ and radius R of minimum volume. Mapped normal examples fall within, whereas mapped anomalies fall outside the hypersphere.

$\boldsymbol{x} \in \mathcal{X}$ given by network ϕ_ω. The aim of Deep SVDD is to jointly learn the network weights ω together with minimizing the volume of a data-enclosing hypersphere in output space $\mathcal{Z}$ that is characterized by radius $R > 0$ and center $\boldsymbol{c} \in \mathcal{Z}$. For now, we assume $\boldsymbol{c} \in \mathcal{Z}$ to be given. Given some training data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$, we define the *soft-boundary Deep SVDD* objective as:

$$\min_{R,\omega} \quad R^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \max\{0, \|\phi_\omega(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - R^2\} + \frac{\lambda}{2} \sum_{l=1}^{L} \|\omega_l\|_{\mathrm{F}}^2. \tag{2.10}$$

As in kernel SVDD, minimizing R^2 minimizes the volume of the hypersphere. The second term penalizes points lying outside the sphere after being passed through the network, that is, if the distance to the center $\|\phi_\omega(\boldsymbol{x}_i) - \boldsymbol{c}\|$ is greater than radius R. Hyperparameter $\nu \in (0, 1]$ controls the trade-off between the volume of the sphere and violations of the boundary, allowing some points to be mapped outside the sphere. In Section 2.2.2, we show that the ν-parameter exactly allows us to control the proportion of outliers in a model similar to the ν-property for kernel methods mentioned previously. The last term is a weight decay regularizer of the network parameters ω with hyperparameter $\lambda > 0$, where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm.

Optimizing objective (2.10) lets the network learn weights ω such that data points are closely mapped to the center $\boldsymbol{c}$ of the hypersphere. To achieve this, the network must extract the common factors of variation of the data. As a result, normal examples of the data are closely mapped to center $\boldsymbol{c}$, whereas anomalous examples are mapped further away from the center or outside of the hypersphere. Minimizing the size of the sphere enforces this learning process. Hence, Deep SVDD learns a concentrated region and thus compact description of the normal class by optimizing its objective.

For cases in which we assume the data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ to be fairly clean, we propose a simplified objective. We define the *One-Class Deep SVDD* objective as

$$\min_{\omega} \quad \frac{1}{n} \sum_{i=1}^{n} \|\phi_\omega(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 + \frac{\lambda}{2} \sum_{l=1}^{L} \|\omega_l\|_{\mathrm{F}}^2. \tag{2.11}$$

One-Class Deep SVDD simply employs a quadratic loss for penalizing the distance of every embedding $\phi_\omega(\boldsymbol{x}_i)$ to center $\boldsymbol{c} \in \mathcal{Z}$. The second term is again a weight decay regularizer with hyperparameter $\lambda > 0$. We can think of One-Class Deep SVDD as also trying to find a hypersphere of minimum volume with center $\boldsymbol{c}$. But unlike in soft-boundary Deep SVDD, where the hypersphere is contracted by directly penalizing the radius and the data embeddings that fall outside the sphere, One-Class Deep SVDD contracts the sphere by simply minimizing the mean squared distance of *all* data embeddings to the center. Again, to closely map the data to center $\boldsymbol{c}$, on average, the neural network must extract common factors of variation. We have found this simplified objective to converge faster and be effective in many situations [466, 469, 471]. In light of the unifying view of anomaly detection we formulate in Chapter 4, we will see that we may interpret One-Class Deep SVDD also as a single-prototype deep clustering method (see Sections 4.2.1 and 4.2.4).

Anomaly Score We define the anomaly score $s : \mathcal{X} \to \mathbb{R}$ for both variants of Deep SVDD as the distance of the embedded data point to the center of the hypersphere:

$$s(\boldsymbol{x}) = \|\phi_{\omega^*}(\boldsymbol{x}) - \boldsymbol{c}\|^2, \tag{2.12}$$

where ω^* are the network weights of a trained model. For soft-boundary Deep SVDD, we can adjust this score by subtracting the radius R^* of the trained model so that anomalies (points with embeddings outside the sphere) have positive scores, whereas inliers have negative scores. The weights ω^* (and R^*) completely characterize a Deep SVDD model and thus no data points must be saved for detection at testing time.

Optimization Both Deep SVDD objectives can be optimized with stochastic gradient descent (SGD) and its variants [306, 179] (e.g., Adam [276]) using backpropagation. Training is carried out until convergence. Using SGD enables Deep SVDD to scale with large datasets as its computational complexity scales linearly in the number of training batches and batches can be processed in parallel (e.g., by using GPUs).

Since the network weights ω and radius R generally live on different scales, using one common SGD learning rate can be inefficient for optimizing soft-boundary Deep SVDD. Instead, we suggest to optimize the network weights ω and radius R alternately in a block coordinate descent fashion. That is, we first train the network weights ω for some $k \in \mathbb{N}$ epochs while the radius R is fixed. Then, after every kth epoch, we solve for radius R given the data embeddings from the latest network update, which can be done analytically (see Proposition 4).

2.2.2 Theoretical Properties of Deep SVDD

We here examine Deep SVDD theoretically. First, we analyze three properties (Propositions 1–3) that (in theory) can lead to trivial, uninformative solutions of Deep SVDD and thus should be accounted for (e.g., by means of regularization, see Section 2.2.3). Afterwards, we prove the ν-property for soft-boundary Deep SVDD.

In the following let $J_{\text{soft}}(R, \omega)$ and $J_{\text{oc}}(\omega)$ be the soft-boundary and One-Class Deep SVDD objective functions, as defined in (2.10) and (2.11), respectively. We first show that including the hypersphere center $\boldsymbol{c} \in \mathcal{Z}$ as a free optimization variable can produce a trivial solution for both (non-regularized) Deep SVDD objectives.

Proposition 1 (Zero weights solution)**.** *Let ω^0 be the set of zero network weights, i.e., $\omega_l = \mathbf{0}$ for every $\omega_l \in \omega^0$. For constant zero weights, the network maps any input to the same output, i.e., $\phi_{\omega^0}(\boldsymbol{x}) =: \boldsymbol{c}^0 \in \mathcal{Z}$ is constant for every $\boldsymbol{x} \in \mathcal{X}$. Then, if $\boldsymbol{c} = \boldsymbol{c}^0$, the optimal solution of Deep SVDD is given by $\omega^* = \omega^0$ and $R^* = 0$.*

Proof. For every parameter configuration (R, ω) we have that both $J_{\text{soft}}(R, \omega) \geq 0$ and $J_{\text{oc}}(\omega) \geq 0$. As the output of the zero weights network $\phi_{\omega^0}(\boldsymbol{x})$ is constant, and the center of the hypersphere is given by $\boldsymbol{c} = \phi_{\omega^0}(\boldsymbol{x})$, all errors in the empirical term of the objectives become zero. Hence, $R^* = 0$ and $\omega^* = \omega^0$ are optimal solutions since we have $J_{\text{soft}}(\omega^*, R^*) = 0$ and $J_{\text{oc}}(\omega^*) = 0$ in this case. □

Proposition 1 implies that if we include the hypersphere center $\boldsymbol{c}$ as an optimization variable, optimizing the (non-regularized) Deep SVDD objectives via SGD may converge to the trivial solution $(R^*, \boldsymbol{c}^*, \omega^*) = (0, \boldsymbol{c}^0, \omega^0)$. We call such a solution, where the network learns a constant map to some fixed output, "hypersphere collapse" since the hypersphere collapses to zero volume. Next, we identify two network architecture properties, which can also encourage such trivial hypersphere collapse solutions.

Proposition 2 (Bias terms)**.** *Let $\boldsymbol{c} \in \mathcal{Z}$ be any fixed hypersphere center. If there is a hidden layer in network $\phi_\omega : \mathcal{X} \to \mathcal{Z}$ having a bias term, there exists an optimal solution (R^*, ω^*) of the Deep SVDD objectives* (2.10) *and* (2.11) *with $R^* = 0$ and $\phi_{\omega^*}(\boldsymbol{x}) = \boldsymbol{c}$ for every $\boldsymbol{x} \in \mathcal{X}$.*

Proof. Assume layer $l \in \{1, \ldots, L\}$ with weights ω_l has a bias term b^l. For any input $\boldsymbol{x} \in \mathcal{X}$, the output of layer l is then given by

$$\boldsymbol{z}^l(\boldsymbol{x}) = \sigma^l(\omega_l \cdot \boldsymbol{z}^{l-1}(\boldsymbol{x}) + b^l),$$

where "$\cdot$" denotes a linear operator (e.g., matrix multiplication or convolution), $\sigma^l(\cdot)$ is the activation function of layer l, and $\boldsymbol{z}^{l-1}(\boldsymbol{x})$ is the output of the previous layer which depends on input $\boldsymbol{x}$. Then, for $\omega_l = \mathbf{0}$, we have that $\boldsymbol{z}^l(\boldsymbol{x}) = \sigma^l(b^l)$, i.e., the output of layer l is constant for every input $\boldsymbol{x} \in \mathcal{X}$. Therefore, the bias term b^l (and the weights of the subsequent layers) can be chosen such that $\phi_{\omega^*}(\boldsymbol{x}) = \boldsymbol{c}$ for every $\boldsymbol{x} \in \mathcal{X}$ (assuming $\boldsymbol{c}$ is in the image of the network as a function of b^l and the weights $\omega_{l+1}, \ldots, \omega_L$ of the subsequent layers). Hence, selecting ω^* in this way results in an empirical term of zero in (2.10) and (2.11), and choosing $R^* = 0$ gives the optimal solution (ignoring the weight decay regularization term for simplicity). □

In other words, Proposition 2 implies that networks with bias terms can easily learn any constant function that is independent of the input $\boldsymbol{x} \in \mathcal{X}$.[2]

[2]Proposition 2 also explains why autoencoders with bias terms are vulnerable to converge to a constant mapping onto the mean, which is the optimal constant solution of the mean squared error.

Proposition 3 (Bounded activation functions)**.** *Consider a network unit having a monotonic activation function $\sigma(\cdot)$ that has an upper (or lower) bound with $\sup_h \sigma(h) \neq 0$ (or $\inf_h \sigma(h) \neq 0$). Then, for a set of unit inputs $\{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n\}$ that have at least one feature that is positive or negative for all inputs, the non-zero supremum (or infimum) can be uniformly approximated on the set of inputs.*

Proof. W.l.o.g. consider the case that σ is upper bounded by $B := \sup_h \sigma(h) \neq 0$ and feature j being positive for all inputs, i.e. $h_{ij} > 0$ for all $i = 1, \ldots, n$. Then, for every $\varepsilon > 0$, one can always choose the jth element w_j of the network unit weights sufficiently large (setting all other network unit weights to zero) such that $\sup_i |\sigma(w_j\, h_{ij}) - B| < \varepsilon$. □

Proposition 3 simply states that a network unit with a monotonic, bounded activation function (e.g., sigmoid or tanh) can be saturated for all inputs, if these inputs share at least one feature with the same sign. Such a saturated unit, by effectively being constant over the inputs, emulates a bias term in the subsequent layer, which again enables the network to more easily learn a constant map (see Proposition 2).

To summarize the above analysis: optimizing the hypersphere center $\boldsymbol{c}$ (due to the zero weights solution) as well as using bias terms and bounded activation functions in the network can foster a trivial hypersphere collapse solution for standard (non-regularized) Deep SVDD. Empirically, we found that using Batch Normalization [248] in the network ϕ_ω (with the top layer excluded, of course), which prevents a collapse in the lower layers due to the normalization, and fixing hypersphere center $\boldsymbol{c}$ as the mean of the data embeddings obtained from an initial forward pass on the training data to be a reasonable strategy for standard (non-regularized) Deep SVDD in practice. We found this strategy, together with the added stochasticity of mini-batch SGD optimization, to produce fairly stable and consistent results, which did not suffer from a hypersphere collapse. In the next Section 2.2.3, we will discuss further regularization techniques, which actively regularize against a trivial collapse solution. But before, we lastly prove that the ν-property also holds for soft-boundary Deep SVDD which allows us to model some target false alarm rate or account for training data contamination.

Proposition 4 (ν-property)**.** *Hyperparameter $\nu \in (0, 1]$ in the soft-boundary Deep SVDD objective* (2.10) *is an upper bound on the fraction of points being outside and a lower bound on the fraction of points being outside or on the boundary of the hypersphere.*

Proof. Define $d_i = \|\phi_\omega(\boldsymbol{x}_i) - \boldsymbol{c}\|^2$ for $i = 1, \ldots, n$. W.l.o.g. assume $d_1 \geq \cdots \geq d_n$. The number of points being outside the hypersphere is given by $n_{\text{out}} = |\{i \mid d_i > R^2\}|$ and we can write the soft-boundary objective J_{soft} (in radius R) as

$$J_{\text{soft}}(R) = R^2 - \frac{n_{\text{out}}}{\nu n} R^2 = \left(1 - \frac{n_{\text{out}}}{\nu n}\right) R^2.$$

That is, radius R should be decreased as long as $n_{\text{out}} \leq \nu n$ holds and decreasing R gradually increases n_{out}. Thus, $\frac{n_{\text{out}}}{n} \leq \nu$ must hold in the optimum, i.e. ν is an

upper bound on the fraction of outliers, and the optimal radius R^* is given for the largest n_{out} for which this inequality still holds. Finally, we have that $R^{*2} = d_i$ for $i = n_{\text{out}} + 1$ since radius R is minimal in this case and points on the boundary do not increase the objective. Hence, we also have $|\{i \mid d_i \geq R^{*2}\}| \geq n_{\text{out}} + 1 \geq \nu n$. □

2.2.3 Regularization and Variants

As illustrated for Deep SVDD above, a recurring question in deep one-class classification is how to meaningfully regularize against a feature map collapse $\phi_\omega \equiv \boldsymbol{c}$, in which case the feature space would be trivially concentrated. To date, several techniques have been proposed in the literature for this. These include adding a reconstruction term or architectural constraints (e.g., removing bias terms) [466, 585], freezing the embedding [155, 468, 409, 423, 269], inversely penalizing the embedding variance [111], using true [411, 469], auxiliary [221, 409, 471, 339], or artificial [339] negative examples in training, pseudo-labeling [181, 222, 111, 52], or integrating some manifold assumption [193]. Note that this question of feature map collapse and its regularization is also raised in deep clustering (see Section 4.2.4 and [66]), where similar objectives have been formulated based on the premise of learning data representations that are also "compact" in some sense. Although various solutions have been introduced at this point, addressing this issue in theoretically principled ways presents an exciting opportunity for future research.

Besides Deep SVDD variants [466, 469, 585, 172], which employ a hypersphere model $f_\theta(\boldsymbol{x}) = \|\phi_\omega(\boldsymbol{x}) - \boldsymbol{c}\|^2 - R^2$, deep OC-SVM variants that employ a linear model $f_\theta(\boldsymbol{x}) = \rho - \langle \phi_\omega(\boldsymbol{x}), \boldsymbol{w} \rangle$ with an explicit neural network feature map $\phi_\omega(\cdot)$ in objective (2.8) have also been proposed [155, 92]. Deep one-class classification variants that have been introduced include multimodal extensions [172, 52], time series adaptations [506], and methods that employ adversarial learning [476, 424, 477] or transfer learning [409, 423]. In Chapter 3, we will introduce two specific variants in detail: FCDD [339], an explainable deep one-class classification method for images, and CVDD [468], a multi-context one-class classification variant applied to characterize text.

Deep one-class classification methods generally offer a greater modeling flexibility than shallow methods 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). The underlying principle of the one-class classification approach—targeting a discriminative one-class boundary in learning—remains unaltered, however, regardless of whether a deep or shallow feature map is used. Figure 2.2 shows a comparison of three canonical one-class classification models (MVE, SVDD, and Deep SVDD) trained on the *Big Moon, Small Moon* toy data set, each using a different feature representation (raw input, kernel, and neural network). In the next section, we present a first quantitative evaluation of Deep SVDD on the MNIST and CIFAR-10 one vs. rest benchmarks.

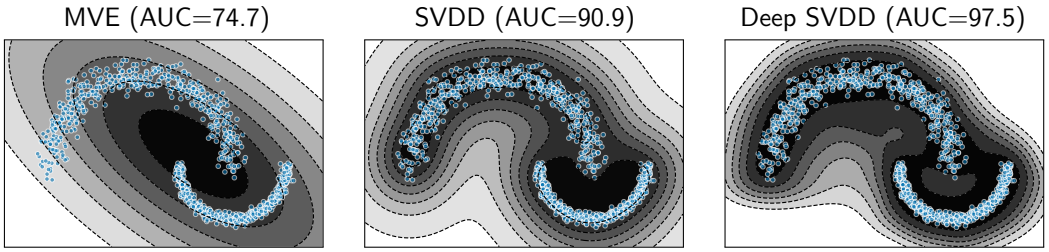


Figure 2.2: One-class classification models on the *Big Moon, Small Moon* toy example from Figure 1.4. A Minimum Volume Ellipsoid (MVE) in input space is limited to enclose an ellipsoidal, convex region. By (implicitly) fitting a hypersphere in kernel feature space, SVDD enables non-convex support estimation. 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.

2.2.4 Experimental Evaluation on MNIST and CIFAR-10

We here evaluate the Deep SVDD method on the well-known MNIST [305] and CIFAR-10 [290] datasets. Images generally provide a good test bed for the usefulness of deep anomaly detection, due to the typically high-dimensional nature of the raw pixel space and the hierarchy of features (from pixels to semantic concepts). Moreover, images allow for an easy qualitative visual assessment of detected anomalies by human observers. Reusing classification datasets to create anomaly detection setups enables a systematic quantitative evaluation of methods, making use of the ground-truth labels available at testing time [153]. As a quantitative evaluation measure, we use the standard AUC measure (see 1.3.3). We compare Deep SVDD against deep and shallow methods from different paradigms.[3]

Setup Both MNIST and CIFAR-10 have ten different classes from which we create ten one vs. rest setups. In each setup, one of the classes is considered as the normal class and samples from the remaining nine classes are used to represent anomalies. This setup in particular allows us to test methods for the detection of semantic anomalies (images of different object classes). We use the original training and test dataset splits and only train on the training set examples from the respective normal class. This results in training set sizes of $n \approx 6\,000$ for MNIST and $n = 5\,000$ for CIFAR-10 in each setup. Both test sets have 10 000 samples and include samples from the respective nine anomalous classes. We rescale pixels to $[0, 1]$ via min-max feature scaling.

Shallow Baselines (i) Kernel OC-SVM/SVDD with Gaussian kernel. We select the inverse length scale γ from $\gamma \in \{2^{-10}, 2^{-9}, \ldots, 2^{-1}\}$ via grid search using the performance on a small holdout set (10 % of randomly drawn test samples). This grants shallow SVDD a small supervised advantage. We run all experiments for

[3] A `PyTorch` implementation of Deep SVDD and code to reproduce the results is available at: `https://github.com/lukasruff/Deep-SVDD-PyTorch`

Table 2.1: Mean AUC (in %) detection performance with std. dev. (over 10 seeds) for various methods on the MNIST and CIFAR-10 one vs. rest benchmarks.

		OC-SVM/ SVDD	KDE	iForest	AE	AnoGAN	Soft DSVDD	One-Class DSVDD
MNIST	0	**98.6**±0.0	97.1±0.0	98.0±0.3	97.6±0.7	96.6±1.3	97.8±0.7	98.0±0.7
	1	99.5±0.0	98.9±0.0	97.3±0.4	98.3±0.6	99.2±0.6	99.6±0.1	**99.7**±0.1
	2	82.5±0.1	79.0±0.0	88.6±0.5	85.4±2.4	85.0±2.9	89.5±1.2	**91.7**±0.8
	3	88.1±0.0	86.2±0.0	89.9±0.4	86.7±0.9	88.7±2.1	90.3±2.1	**91.9**±1.5
	4	**94.9**±0.0	87.9±0.0	92.7±0.6	86.5±2.0	89.4±1.3	93.8±1.5	**94.9**±0.8
	5	77.1±0.0	73.8±0.0	85.5±0.8	78.2±2.7	88.3±2.9	85.8±2.5	**88.5**±0.9
	6	96.5±0.0	87.6±0.0	95.6±0.3	94.6±0.5	94.7±2.7	98.0±0.4	**98.3**±0.5
	7	93.7±0.0	91.4±0.0	92.0±0.4	92.3±1.0	93.5±1.8	92.7±1.4	**94.6**±0.9
	8	88.9±0.0	79.2±0.0	89.9±0.4	86.5±1.6	84.9±2.1	92.9±1.4	**93.9**±1.6
	9	93.1±0.0	88.2±0.0	93.5±0.3	90.4±1.8	92.4±1.1	94.9±0.6	**96.5**±0.3
CIFAR-10	airplane	61.6±0.9	61.2±0.0	60.1±0.7	59.1±5.1	**67.1**±2.5	61.7±4.2	61.7±4.1
	automobile	63.8±0.6	64.0±0.0	50.8±0.6	57.4±2.9	54.7±3.4	64.8±1.4	**65.9**±2.1
	bird	50.0±0.5	50.1±0.0	49.2±0.4	48.9±2.4	**52.9**±3.0	49.5±1.4	50.8±0.8
	cat	55.9±1.3	56.4±0.0	55.1±0.4	58.4±1.2	54.5±1.9	56.0±1.1	**59.1**±1.4
	deer	66.0±0.7	**66.2**±0.0	49.8±0.4	54.0±1.3	65.1±3.2	59.1±1.1	60.9±1.1
	dog	62.4±0.8	62.4±0.0	58.5±0.4	62.2±1.8	60.3±2.6	62.1±2.4	**65.7**±2.5
	frog	74.7±0.3	**74.9**±0.0	42.9±0.6	51.2±5.2	58.5±1.4	67.8±2.4	67.7±2.6
	horse	62.6±0.6	62.6±0.0	55.1±0.7	58.6±2.9	62.5±0.8	65.2±1.0	**67.3**±0.9
	ship	74.9±0.4	75.1±0.0	74.2±0.6	**76.8**±1.4	75.8±4.1	75.6±1.7	75.9±1.2
	truck	75.9±0.3	**76.0**±0.0	58.9±0.7	67.3±3.0	66.5±2.8	71.0±1.1	73.1±1.2

$\nu \in \{0.01, 0.1\}$ and report the better result. (ii) Kernel Density Estimation (KDE). We select the bandwidth h of the Gaussian kernel from $h \in \{2^{0.5}, 2^1, \ldots, 2^5\}$ via 5-fold cross-validation using the log-likelihood score. (iii) Isolation Forest (iForest). We set the number of trees to $t = 100$ and the sub-sampling size to $\psi = 256$, as recommended by the authors [334]. For all three shallow baselines, we reduce the dimensionality of the data via PCA, where we choose the a minimum number of principal components such that at least 95% of the variance is retained [155].

Deep Competitors We compare Deep SVDD to a reconstruction-based convolutional autoencoder (AE) and the generative AnoGAN [488] method. For the AE encoder, we use exactly the same networks as for Deep SVDD. We then create the respective decoders symmetrically, where we substitute convolutions with deconvolutions and max-pooling with upsampling respectively. For AnoGAN we use the DCGAN architecture [439] and set the latent space dimensionality to 256, following Metz et al. [368], and otherwise follow Schlegl et al. [488]. For Deep SVDD, we set the hypersphere center $\boldsymbol{c}$ to the mean of the embedded data after performing an initial forward pass. We initialize the Deep SVDD network with the weights from the pre-trained AE. For soft-boundary Deep SVDD, we choose ν from $\nu \in \{0.01, 0.1\}$ and again report the better result. We use the Adam optimizer with standard parameters [276] and apply Batch Normalization [248] in the networks. We employ a simple

two-phase learning rate schedule with an initial learning rate of $\eta = 10^{-4}$ followed by $\eta = 10^{-5}$. We train the AE for 250 + 100 and Deep SVDD for 150 + 100 epochs.

Network Architectures For both datasets, we use LeNet-type convolutional neural networks (CNNs), where each network module consists of a convolutional layer followed by leaky ReLU activations (with $\alpha = 0.1$) and 2×2 max-pooling. On MNIST, we use a CNN with two modules, $8 \times (5 \times 5 \times 1)$-filters followed by $4 \times (5 \times 5 \times 1)$-filters, and a final fully connected layer with 32 units. On CIFAR-10, we use a CNN with three modules, $32 \times (5 \times 5 \times 3)$-filters, $64 \times (5 \times 5 \times 3)$-filters, and $128 \times (5 \times 5 \times 3)$-filters, followed by a final fully connected layer with 128 units. We use a batch size of 200 and set the weight decay hyperparameter to $\lambda = 10^{-6}$.

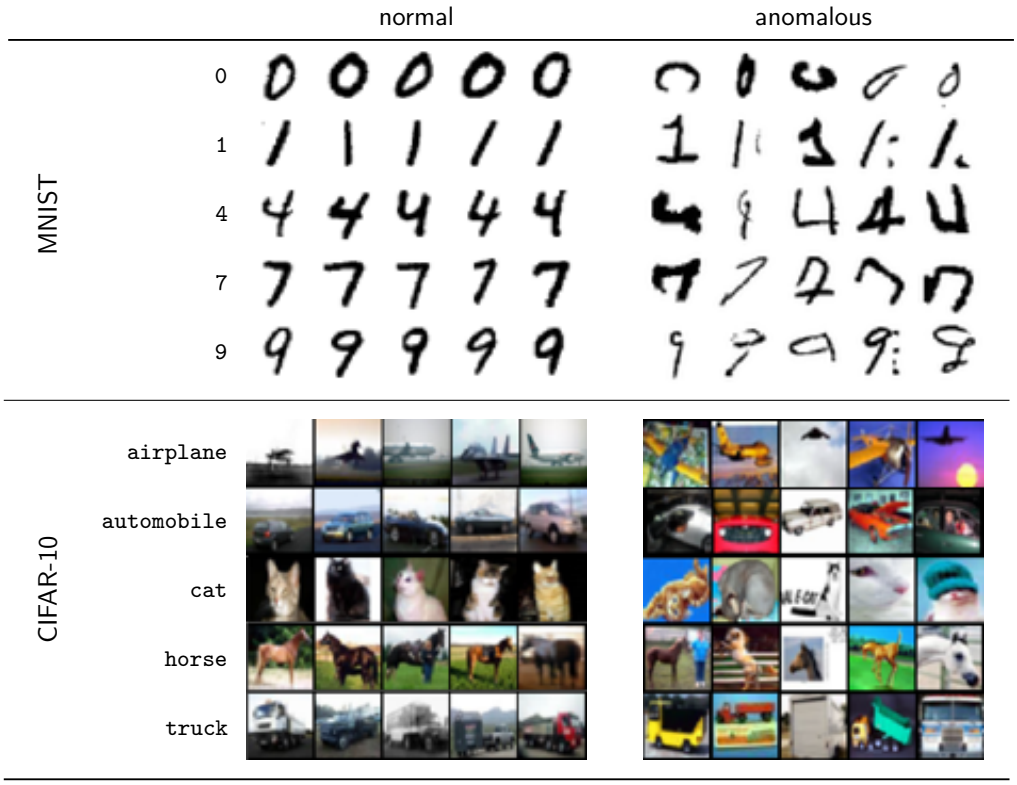


Figure 2.3: Most normal (left) and most anomalous (right) in-class examples determined by One-Class Deep SVDD for selected MNIST and CIFAR-10 one-class experiments. We can see that Deep SVDD correctly assigns high anomaly scores to images with unusual shapes, colors, or patterns.

Results and Discussion The results are presented in Table 2.1. Deep SVDD shows the best performance on the MNIST benchmark. On CIFAR-10, the results are more mixed. We can observe that One-Class Deep SVDD consistently performs slightly better than the soft-boundary counterpart on both datasets. One reason for this is

probably that in the one vs. rest benchmark we have relatively clean training data (only containing images of the respective normal class). Another reason might be that the smooth L^2 loss of One-Class Deep SVDD may be better for SGD optimization, as the output gradients are more informative than the gradients of the soft-boundary maximum rectifier. In Figure 2.3, we show some examples of the most normal and most anomalous in-class images according to (One-Class) Deep SVDD, where we can see that Deep SVDD correctly detects examples with unusual shapes and patterns. The anomalous images also show that there are some unusual in-class variations that can be considered as noise or pollution. Moreover, note that the shallow SVDD and KDE perform better than all the deep methods on three out of the ten CIFAR-10 classes (`deer`, `frog`, and `truck`). Figure 2.4 displays examples of the most normal and most anomalous in-class images according to KDE respectively. Here, we can see that the normal images of the three classes on which KDE performs best seem to have a pronounced global structure: the `truck` images are mostly divided horizontally into street and sky, and `deer` as well as `frog` globally have similar uniform colors. For these classes, choosing local CNN features may be questioned.

Figure 2.4: Most normal (left) and most anomalous (right) in-class examples determined by KDE for CIFAR-10 one-class experiments in which KDE performs best. We can observe that the normal images seem to be dominated by similar global structures (e.g., green background for `deer` or a horizontal split into street and sky for `truck`)

2.3 Deep Semi-Supervised One-Class Classification

Unlike the standard unsupervised anomaly detection setting, in many real-world applications one may also have access to some verified (i.e., labeled) normal or anomalous samples in addition to the unlabeled data. Such samples could be identified and labeled by a domain expert, for instance. This leads to a semi-supervised anomaly detection setting, as we have specified in Section 1.3.2 in the introduction, where we have $n \in \mathbb{N}$ (mostly normal but possibly containing some anomalous contamination) unlabeled samples and $m \in \mathbb{N}$ labeled samples,

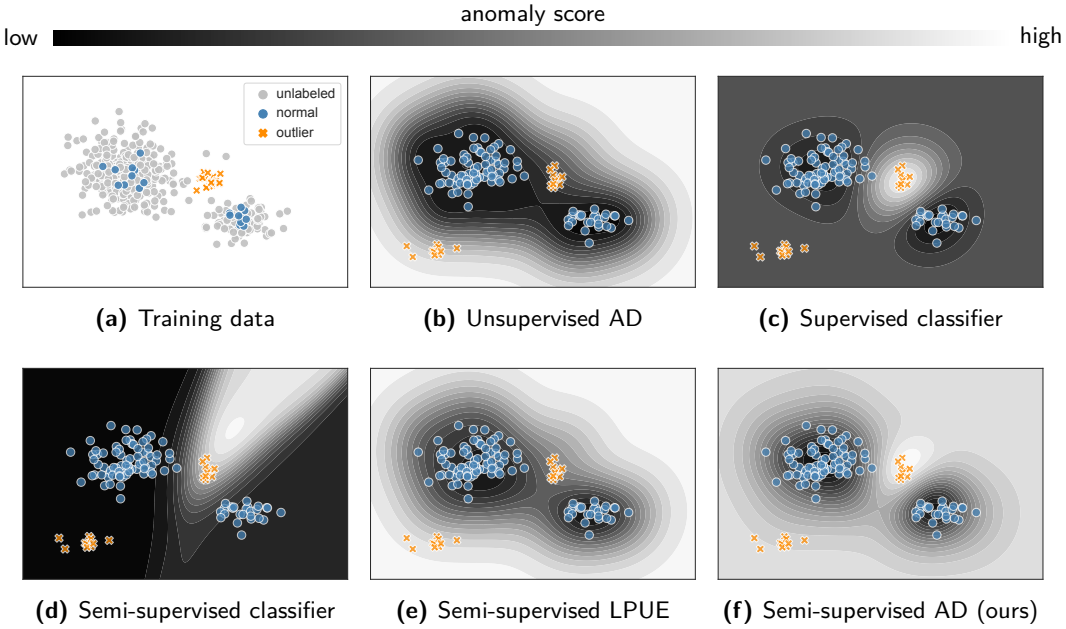
$$\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X} \quad \text{and} \quad (\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \dots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y},$$

where $\tilde{y} = +1$ and $\tilde{y} = -1$ denote normal and anomalous samples respectively.

As mentioned in Section 1.3.2, the term "semi-supervised anomaly detection" has been used to describe two different anomaly detection settings. Most existing "semi-supervised" anomaly detection methods, both shallow [95, 384, 63] and deep [522, 13, 90], only incorporate the use of labeled normal samples but not labeled anomalies, that is, they are more precisely instances of Learning from Positive (i.e., normal) and Unlabeled Examples (LPUE) [134, 607, 145]. A few works [573, 338, 191] have also investigated the semi-supervised setting where one also utilizes labeled anomalies. However, previous deep approaches have been mostly domain or data-type specific [156, 280, 371].

Research on deep semi-supervised learning in general has mostly focused on classification as the downstream task [279, 444, 400, 122, 403]. Such semi-supervised classifiers typically assume that similar points are likely of the same class and that there is a closed set of classes, with each class having some distinct structure (e.g., being clustered or concentrated). This assumption, however, only holds for the "normal class" in anomaly detection (see *concentration assumption* in Section 1.3.1), but is critically invalid for the "anomaly class" since anomalies are not necessarily similar to one another. Instead, semi-supervised anomaly detection approaches must find a compact description of the normal class while also correctly discriminating the labeled anomalies [191]. We illustrate the ramifications of the various learning paradigms for anomaly detection on a toy example in Figure 2.5.

In this section, we introduce Deep SAD, a deep one-class classification method for semi-supervised anomaly detection. We will give an information-theoretic motivation for the method formulation, and will experimentally evaluate the use of Deep SAD first in settings where few true anomalies are available. Afterwards, in contrast to having few true anomalies available, we will look at the situation where plenty "auxiliary anomalies" are available from some large unlabeled corpus (e.g., 80 Million Tiny Images [548]). In this evaluation, we will also look at Deep SVDD and Deep SAD from another perspective, namely through the lens of a binary cross-entropy objective with probabilities modeled in terms of radial basis functions. But first,



(a) Training data **(b)** Unsupervised AD **(c)** Supervised classifier

(d) Semi-supervised classifier **(e)** Semi-supervised LPUE **(f)** Semi-supervised AD (ours)

Figure 2.5: The need for semi-supervised anomaly detection: The training data (shown in (a)) consists of (mostly normal) unlabeled data (gray) as well as a few labeled normal samples (blue) and labeled anomalies (orange). The ground-truth normal law $\mathbb{P}^+$ is a mixture of two Gaussians. Figures (b)–(f) show the decision boundaries for models of the various learning paradigms at testing time when novel anomalies occur (bottom left in each plot).
An unsupervised anomaly detection (AD) model (b) does not take advantage of labeled training samples. A supervised classifier (c) overfits to the anomalies seen at training and fails to generalize to the novel anomalies. A semi-supervised classifier (d) makes a binary cluster assumption and also fails to generalize to the novel anomalies. Learning from positive and unlabeled examples (LPUE) (e) improves over an unsupervised model using the labeled normal training samples, but ignores the known anomalies. Our semi-supervised AD approach (f) takes advantage of all training data: unlabeled samples, labeled normal samples, as well as labeled anomalies. This strikes a balance between one-class learning and classification.

we will briefly discuss the different qualitative types of negative examples that exist next.

2.3.1 Negative Examples

We recognize three qualitative types of negative examples that have been studied in the literature, that we distinguish as "artificial," "auxiliary," and "true" negative examples which 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 Section 14.2.4 in Hastie et al. [214] for example). 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 [525]. 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 [524] such as resampling schemes [540], manifold sampling [128], and sampling based on local density estimation [159, 102] as well as active learning strategies [4, 526, 190]. In Section 3.1, we will use an artificial sampling scheme with the FCDD method to improve anomaly localization for visual inspection in manufacturing.

A more recent idea is to treat the enormous quantities of data that are publicly available in some domains as *auxiliary* negative examples [221], 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 simply sampling random pixels or words. As previously mentioned, this approach is also known as Outlier Exposure (OE) [221]. Note again, that such auxiliary examples may not coincide with anomalies at testing time, but they can nevertheless be informative to improve detection performance in some domains (see Section 2.3.6 and Section 3.1.2 as well as [221, 222, 471]). OE has also been used with density-based methods (see Section 4.1) by employing a margin loss [221] or using temperature annealing [487] on the log-likelihood ratio between positive and negative examples.

The most informative labeled negative examples are ultimately *true* anomalies, for example data points verified as anomalous by some domain expert in a specific application. Including even a few labeled anomalies can significantly improve the detection performance of a model, as has been found for kernel methods [536, 191] and which we will confirm for deep methods below. There also have been active learning algorithms proposed that include subjective user feedback (e.g., from an expert) to learn about the user-specific informativeness of particular anomalies in an application [421]. We finally remark that negative examples have also been heuristically incorporated into reconstruction models (see Section 4.2), by employing a *bounded* reconstruction error [144], since maximizing the unbounded error for negative examples quickly becomes unstable.

2.3.2 The Deep SAD Method

We here introduce an end-to-end deep method for Semi-supervised Anomaly Detection (SAD), which we call the Deep SAD method. Let $\phi_\omega : \mathcal{X} \to \mathcal{Z}$ be a neural network with input space $\mathcal{X} \subseteq \mathbb{R}^D$, output space $\mathcal{Z} \subseteq \mathbb{R}^d$, and network weights ω. Given n unlabeled samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X}$ and m labeled samples $(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \dots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y}$ with $\mathcal{Y} = \{\pm 1\}$ where $\tilde{y} = +1$ denotes known normal samples and $\tilde{y} = -1$ known anomalies respectively, we define the *Deep SAD* objective as:

$$\min_{\omega} \quad \frac{1}{n+m} \sum_{i=1}^{n} \|\phi_\omega(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 + \frac{\eta}{n+m} \sum_{j=1}^{m} \left(\|\phi_\omega(\tilde{\boldsymbol{x}}_j) - \boldsymbol{c}\|^2\right)^{\tilde{y}_j} + \frac{\lambda}{2} \sum_{l=1}^{L} \|\omega_l\|_{\mathrm{F}}^2. \quad (2.13)$$

That is, we use the Deep SVDD loss (see (2.11)) for the unlabeled data in the Deep SAD objective, and thus recover Deep SVDD as the special case when there is no labeled training data available ($m = 0$). This again incorporates the assumption that most of the unlabeled data is assumed to be normal (which may be adapted).

For the labeled data, we introduce a new loss term that is weighted with a hyperparameter $\eta > 0$, which controls the balance between the labeled and the unlabeled term. Setting $\eta > 1$ puts more emphasis on the labeled data whereas $\eta < 1$ emphasizes the unlabeled data. For the labeled normal samples ($\tilde{y} = +1$), we also impose a quadratic loss on the distances of the mapped points to the center $\boldsymbol{c}$, thus intending to learn a latent distribution that overall concentrates the normal data. One might consider $\eta > 1$ here to emphasize the labeled over unlabeled normal samples (e.g., to reflect labeling confidence). For the labeled anomalies ($\tilde{y} = -1$), in contrast, we penalize the *inverse* of the distances such that anomalies must be mapped further away from the center.[4] Note that this loss formulation is in line with the assumption that anomalies are not concentrated (see *concentration assumption* in Section 1.3.1). In our experiments, we found $\eta = 1$ to be a good default choice, which yielded consistent results. We present a sensitivity analysis on η in the experimental evaluation in Section 2.3.4. As for Deep SVDD (in (2.12)), we define the Deep SAD anomaly score as the distance of a mapped point to the center $\boldsymbol{c}$, i.e. $s(\boldsymbol{x}) = \|\phi_\omega(\boldsymbol{x}) - \boldsymbol{c}\|$, and optimize the Deep SAD objective (2.13) via SGD using backpropagation.

In addition to the inverse squared norm loss, we experimented with several other losses including the negative squared norm loss, negative robust losses, and the hinge loss. The negative squared norm loss is unbounded from below and results in an ill-posed optimization problem causing optimization to diverge. Negative robust losses, such as the Hampel loss [210], introduce one or more scale parameters which are difficult to select or optimize in conjunction with the changing representation learned by the network. Similar to our experimental evaluation in 2.2.4, we found that using a hinge loss consistently resulted in a slightly poorer performance (see also [467]). The inverse squared norm loss instead is bounded from below and smooth, which are crucial properties for losses used in deep learning [187], and ultimately performed the best while remaining conceptually simple. In Section 2.3.5, we present another loss based on using radial basis functions with cross-entropy, which is similar in shape, but different in rate to the Deep SAD loss.

2.3.3 An Information-Theoretic View

The study of the theoretical foundations of deep learning is an active and ongoing research effort [376, 544, 117, 152, 392, 440, 608, 5, 29, 45, 579, 299]. One important line of research that has emerged is rooted in information theory [504]. In supervised deep learning, where one has an input variable X, a latent variable Z (e.g., the final layer of a deep network), and an output variable Y (i.e., the label), the well-known *Information Bottleneck* principle [545, 544, 510, 16, 484] provides an explanation for

[4] To ensure numerical stability, we add a machine epsilon ($\texttt{eps} \sim 10^{-6}$) to the denominator.

representation learning as the trade-off between finding a minimal compression Z of the input X while retaining the informativeness of Z for predicting the label Y. Put formally, supervised deep learning seeks to minimize the mutual information $\mathcal{I}(X;Z)$ between the input X and the latent representation Z while maximizing the mutual information $\mathcal{I}(Z;Y)$ between Z and the classification task Y, i.e.

$$\min_{p(\boldsymbol{z}\,|\,\boldsymbol{x})} \quad \mathcal{I}(X;Z) - \alpha\,\mathcal{I}(Z;Y), \tag{2.14}$$

where $p(\boldsymbol{z}\,|\,\boldsymbol{x})$ is modeled by a deep network and the hyperparameter $\alpha > 0$ controls the trade-off between compression (i.e., complexity) and classification accuracy.

For unsupervised deep learning, due to the absence of labels Y and thus the lack of a clear learning task, other information-theoretic learning principles have been formulated. Of these, the *Infomax* principle [330, 46, 231] is one of the most prevalent and widely used principles. In contrast to (2.14), the objective of Infomax is to *maximize* the mutual information $\mathcal{I}(X;Z)$ between the data X and its latent representation Z:

$$\max_{p(\boldsymbol{z}\,|\,\boldsymbol{x})} \quad \mathcal{I}(X;Z) + \beta\,\mathcal{R}(Z). \tag{2.15}$$

This is typically done under some additional constraint or regularization $\mathcal{R}(Z)$ on the representation Z with hyperparameter $\beta > 0$ to obtain some properties desired for a specific downstream task. Examples where the Infomax principle has been applied include tasks such as independent component analysis [46], clustering [519, 254], generative modeling [107, 616, 15], and unsupervised representation learning in general [231].

We note that the Infomax principle also appears in existing deep anomaly detection methods. Most notably, autoencoders [472, 229], which constitute one major approach to deep anomaly detection (see Section 4.2), can be understood as implicitly maximizing the mutual information $\mathcal{I}(X;Z)$ via the reconstruction objective [565] under some regularization of the latent code Z. Choices for regularization include sparsity [352], the distance to some latent prior distribution (e.g., measured via the KL divergence [277, 454, 278]), adversarial training [353], or simply a bottleneck in dimensionality. Such restrictions share the idea that the latent representation of the normal data should be in some sense "compact."

We now observe that Deep SVDD may not only be interpreted in geometric terms as following minimum volume estimation [432, 500], but also in probabilistic terms as following an entropy minimization over the latent distribution. For a latent random variable Z with covariance Σ, pdf $p(\boldsymbol{z})$, and support $\mathcal{Z} \subseteq \mathbb{R}^d$, we have the following bound on the entropy $\mathcal{H}(Z)$ of Z:

$$\mathcal{H}(Z) = \mathbb{E}[-\log p(Z)] = -\int_{\mathcal{Z}} p(\boldsymbol{z}) \log p(\boldsymbol{z})\ \mathrm{d}\boldsymbol{z} \leq \frac{1}{2}\log((2\pi e)^d \det \Sigma), \tag{2.16}$$

which holds with equality iff Z is jointly Gaussian [120]. Assuming that the latent distribution Z follows an isotropic Gaussian, $Z \sim N(\boldsymbol{c}, \sigma^2 I)$ with $\sigma > 0$, we get

$$\mathcal{H}(Z) = \frac{1}{2}\log((2\pi e)^d \det \sigma^2 I) = \frac{d}{2}(1 + \log(2\pi\sigma^2)) \propto \log \sigma^2, \tag{2.17}$$

i.e. for a fixed dimensionality d, the entropy of Z is proportional to its log-variance.

Now recall that the One-Class Deep SVDD objective (2.11) is to closely map the normal data to some center $\boldsymbol{c} \in \mathcal{Z}$ in latent space $\mathcal{Z} \subseteq \mathbb{R}^d$, measured in squared L^2-norm distance. This is equivalent to minimizing the empirical variance in latent space (disregarding weight decay regularization) and thus minimizing an upper bound on the entropy of a latent Gaussian. When pre-training the Deep SVDD network on an autoencoding objective, which implicitly maximizes the mutual information $\mathcal{I}(X; Z)$ [565], we thus may interpret Deep SVDD as following the Infomax principle (2.15) with an additional "concentration regularization" that the latent distribution should have minimal entropy. Following this, we can view the Deep SAD objective as modeling the latent distribution of normal data to have *low entropy*, and the latent distribution of anomalies to have *high entropy*, as the inverse squared L^2-norm loss for labeled anomalies induces a latent distribution with high entropy for the anomalous data. This formulation does notably not impose any concentration assumption on the anomaly distribution $\mathbb{P}^-$ (see Section 1.3.1).

Similarly, starting from an information-theoretic perspective on the β-VAE, which can be interpreted as making a rate-distortion trade-off [51] between balancing the latent compression (negative rate) and reconstruction accuracy (distortion) [225, 15], Park et al. [415] have recently established a connection between Deep SVDD and VAEs. Their analysis shows that Deep SVDD can be seen as a special case of β-VAE that only seeks to minimize the rate (i.e., maximize compression) of the normal data. Both of these views offer potentially interesting starting points for future theoretical analysis and insight (see also Section 5.2.6).

2.3.4 Experimental Evaluation on Using Few True Anomalies

We evaluate Deep SAD on the MNIST, Fashion-MNIST, and CIFAR-10 datasets as well as on classic anomaly detection benchmarks, creating scenarios where few ground-truth anomalies are available for training. In our experiments, we make a comparison to shallow, hybrid, and deep unsupervised, semi-supervised, as well as supervised competitors.[5]

Competitors We consider the OC-SVM/SVDD with a Gaussian kernel (which in this case are equivalent), Isolation Forest [334], and KDE [417] as shallow unsupervised baselines. For deep unsupervised competitors, we consider (convolutional) autoencoders and the unsupervised Deep SVDD method. For semi-supervised anomaly detection approaches that also take advantage of labeled anomalies, we consider the shallow SSAD method [191] with a Gaussian kernel, which is a semi-supervised extension of kernel SVDD [538]. We further introduce a hybrid SSAD baseline that applies SSAD to the latent codes of the trained autoencoder models. Such hybrid methods acting on deep feature spaces have demonstrated to achieve performance improvements over their raw feature counterparts on high-dimensional data [155, 83, 53].

[5] A `PyTorch` implementation of Deep SAD and code to reproduce the results is available at: `https://github.com/lukasruff/Deep-SAD-PyTorch`

We also include such hybrid variants for all shallow unsupervised baselines. To also compare to a deep semi-supervised learning method that targets classification as the downstream task, we add the well-known Semi-Supervised Deep Generative Model (SS-DGM) [279] where we use the latent class probability estimate (normal vs. anomalous) as the anomaly score. To complete the full learning spectrum, we also include a fully supervised deep classifier trained on the binary cross-entropy loss.

In our experiments we deliberately grant the shallow and hybrid methods an advantage by selecting their hyperparameters to maximize the AUC on a subset (10%) of the test set to mitigate hyperparameter selection issues. To control for architectural effects between the deep methods, we always use the same (LeNet-type) deep networks. We provide the complete network architectures and hyperparameter selection details in Appendix B.1. Since this evaluation includes many methods, we only report the results of methods that have shown to be competitive here, and provide the complete results in Appendix C.2.

Experimental Scenarios on MNIST, Fashion-MNIST, and CIFAR-10

Setup MNIST, Fashion-MNIST, and CIFAR-10 all have ten classes for which we again derive one vs. rest anomaly detection setups on each dataset following many existing works [466, 130, 181, 13, 1, 424, 574, 222, 52]. That is, we set one of the ten classes to be the normal class and let the remaining nine classes represent anomalies in every setup. We use the original training data of the respective normal class as the unlabeled part of our training set. Thus we start with a clean setting that fulfills the assumption that most (in this case all) unlabeled samples are normal. This leads to unlabeled training data sizes of $n \approx 6\,000$ for MNIST and Fashion-MNIST, and $n = 5\,000$ for CIFAR-10 per setup. The training data of the respective nine anomaly classes then forms the data pool from which we draw anomalies for training to create different scenarios. We again compute the AUC on the respective original test sets using ground-truth labels to make a quantitative comparison. We rescale pixels to $[0, 1]$ via min-max feature scaling as the only data pre-processing step.

Experimental Scenarios We examine three scenarios in which we vary the following three experimental parameters: (i) the ratio of labeled training data γ_l, (ii) the pollution ratio γ_p in the unlabeled training data with (unknown) anomalies, and (iii) the number of anomaly classes k_l included in the labeled training data.

(i) Adding Labeled Anomalies In this scenario, we investigate the effect that including labeled true anomalies during training has on detection performance to see the benefit of a semi-supervised anomaly detection approach over other paradigms. For this, we increase the ratio of labeled training data $\gamma_l = \frac{m}{n+m}$ by adding more and more known anomalies $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_m$ with $\tilde{y}_j = -1$ to the training set. The labeled anomalies are always sampled only from one of the nine anomaly classes ($k_l = 1$) per run. For testing, we then consider all nine remaining classes as anomalies, i.e. there are eight novel classes at testing time. We do this to simulate the heterogeneous

nature of anomalies. For the unlabeled part of the training set, we keep the training data of the respective normal class, which we leave unpolluted in this experimental setup ($\gamma_p = 0$). We iterate this training set generation process per setup always over all the nine respective anomaly classes and report the average results over the ten setups $\times$ nine anomaly classes, i.e. over 90 experiments per labeled ratio γ_l.

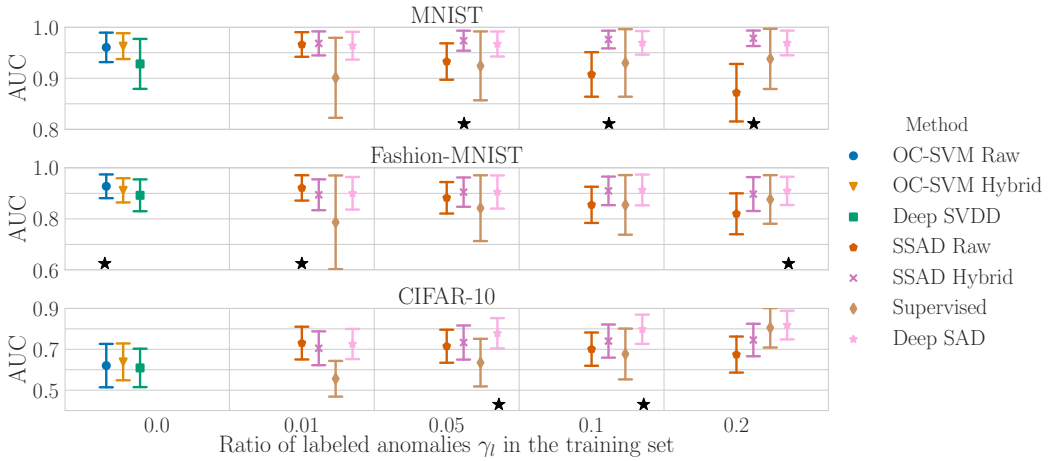


Figure 2.6: Results of scenario (i), where we increase the ratio of labeled anomalies γ_l in the training set. We report mean AUC with std. dev. over 90 experiments for various γ_l. A "$\star$" indicates a statistically significant ($\alpha = 0.05$) difference between the 1st and 2nd best method.

(ii) Polluted Training Data Here we investigate the robustness of the different methods to an increasing pollution ratio γ_p of the training set with unlabeled anomalies. To do so, we pollute the unlabeled part of the training set with anomalies drawn from all nine respective anomaly classes in each setup. We fix the ratio of labeled training samples at $\gamma_l = 0.05$, where we again draw samples only from $k_l = 1$ anomaly class in this scenario. We repeat this training set generation process per setup over all the nine respective anomaly classes and report the average results over the resulting 90 experiments per pollution ratio γ_p. We hypothesize that learning from labeled anomalies in a semi-supervised anomaly detection approach alleviates the negative impact pollution has on detection performance since similar unknown anomalies in the unlabeled data might be detected.

(iii) Number of Known Anomaly Classes In the last scenario, we compare the detection performance at various numbers of known anomaly classes. In scenarios (i) and (ii), we always sample labeled anomalies only from one out of the nine anomaly classes ($k_l = 1$). In this scenario, we now increase the number of anomaly classes k_l included in the labeled part of the training set. Since we have a limited number of anomaly classes (nine) in each setup, we expect the supervised classifier to catch up at some point. We fix the overall ratio of labeled training examples again at

$\gamma_l = 0.05$ and consider a pollution ratio of $\gamma_p = 0.1$ for the unlabeled training data in this scenario. We repeat this training set generation process for ten seeds in each of the ten setups and report the average results over the resulting 100 experiments per number k_l. For each seed, the k_l classes are drawn uniformly at random from the nine respective anomaly classes.

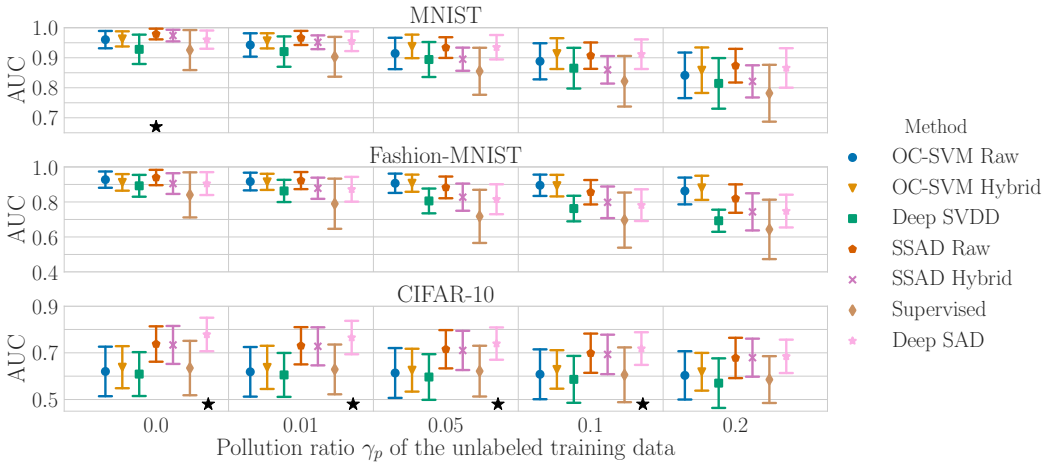


Figure 2.7: Results of scenario (ii), where we pollute the unlabeled part of the training set with (unknown) anomalies. We report mean AUC with std. dev. over 90 experiments for various γ_p. A "$\star$" indicates a statistically significant ($\alpha = 0.05$) difference between the 1st and 2nd best method.

Results and Discussion The results of scenarios (i)–(iii) are shown in Figures 2.6–2.8. In Figure 2.6, we see the advantage of using few labeled anomalies in a deep semi-supervised approach, especially on the most complex CIFAR-10 dataset, where Deep SAD performs the most favorable. On the less complex MNIST and Fashion-MNIST datasets, the unsupervised detectors already establish a strong baseline. Figure 2.6 moreover confirms that a supervised classification approach is vulnerable to novel anomalies at testing time when only little labeled training data is available. In comparison, Deep SAD seems to generalize to novel anomalies while also taking advantage of the labeled examples. Figure 2.7 shows that the detection performance of all methods decreases with increasing data pollution. Deep SAD proves to be the most robust again on CIFAR-10. Interestingly, the unsupervised detectors seem to be more robust on MNIST and Fashion-MNIST, questioning the hypothesis that semi-supervision helps to improve robustness towards pollution. One reason for this might be that contrasting the polluted (but assumed to be normal) unlabeled data with few labeled anomalies, may unfavorably push the decision boundary to let the (unknown) anomalies contained in the pollution appear more normal. Figure 2.8 shows that the more diverse the labeled anomalies in the training set, the better the detection performance becomes. This confirms the natural intuition that the more diverse a set of labeled anomalies is, the more informative these samples are

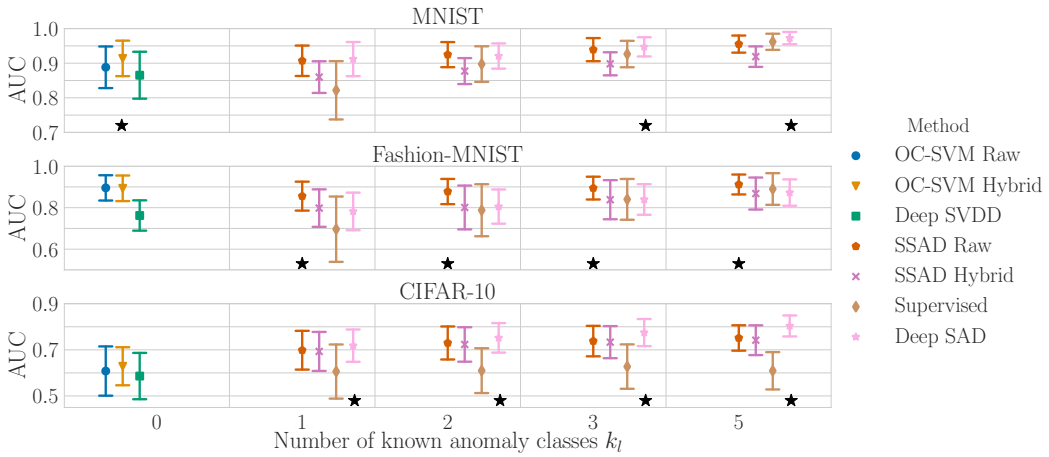


Figure 2.8: Results of scenario (iii), where we increase the number of anomaly classes k_l included in the labeled training data. We report mean AUC with std. dev. over 100 experiments for various k_l. A "$\star$" indicates a statistically significant ($\alpha = 0.05$) difference between the 1st and 2nd best method.

for improving detection. Diversity in the labeled anomalies certainly also helps to improve generalization. We can further see that the supervised method is very sensitive to the number of anomaly classes but catches up at some point as suspected. This does not occur with CIFAR-10, however, where $\gamma_l = 0.05$ labeled training samples seems to be insufficient for classification. Overall, we see that Deep SAD is particularly beneficial on the more complex CIFAR-10 (we provide additional AUC scatterplots in Appendix C.1).

Sensitivity Analysis

We run Deep SAD experiments on the ten one vs. rest setups described above on each dataset for $\eta \in \{10^{-2}, \ldots, 10^{2}\}$ to analyze the sensitivity of Deep SAD with respect to the hyperparameter $\eta > 0$. In this analysis, we set the experimental parameters to a default of $\gamma_l = 0.05$, $\gamma_p = 0.1$, and $k_l = 1$, and again iterate over all nine anomaly classes in every setup. The results shown in Figure 2.9 suggest that Deep SAD is fairly robust against changes of the hyperparameter η. We also provide a sensitivity analysis on varying the output dimension d in Appendix A.1, where we observe that the detection performance increases and

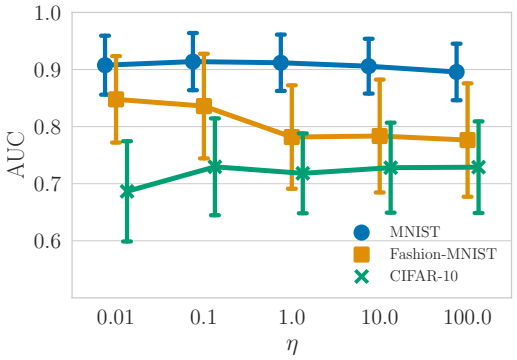


Figure 2.9: Deep SAD sensitivity analysis w.r.t. hyperparameter η. We report mean AUC with std. dev. over 90 experiments for various η.

saturates with increasing dimensionality d.

Classic Anomaly Detection Benchmark Datasets

In this experiment, we examine the detection performance of the various methods on some well-established anomaly detection benchmark datasets [447] listed in Table 2.2. We run these experiments to evaluate the deep versus the shallow approaches also on non-image datasets that are rarely considered in the deep anomaly detection literature.

Table 2.2: Classic benchmark datasets.

Dataset	$n+m$	D	#outliers (%)
arrhythmia	452	274	66 (14.6%)
cardio	1 831	21	176 (9.6%)
satellite	6 435	36	2 036 (31.6%)
satimage-2	5 803	36	71 (1.2%)
shuttle	49 097	9	3 511 (7.2%)
thyroid	3 772	6	93 (2.5%)

Setup We consider random train-to-test set splits of 60:40 while maintaining the original proportion of anomalies in each set. We then run experiments for 10 seeds with $\gamma_l = 0.01$ and $\gamma_p = 0$, i.e. 1% of the training set are labeled anomalies and the unlabeled training data is unpolluted. Since there are no distinct anomaly classes in these datasets, we have $k_l = 1$. We standardize features to have zero mean and unit variance as the only pre-processing step.

Results and Discussion Table 2.3 shows the results, again for the competitive methods. We observe that the shallow kernel methods seem to perform slightly better on the rather small, low-dimensional datasets. Note that Deep SAD also shows competitive results and that the small differences may be explained by the advantage we grant the shallow methods in the selection of their hyperparameters. Another interesting observation in Table 2.3 is that the shallow models trained in the original ("raw") feature space seem to systematically outperform their hybrid variants, which are trained on autoencoder embeddings. This highlights that encoder-induced feature spaces, especially in lower dimensional settings, can also mask information that is relevant for detecting anomalies. We provide the complete results in Appendix C.2.

Table 2.3: Results on classic anomaly detection benchmark datasets in the setting with no pollution $\gamma_p = 0$ and a ratio of labeled anomalies of $\gamma_l = 0.01$ in the training set. We report mean AUC with std. dev. computed over 10 seeds.

	OC-SVM Raw	OC-SVM Hybrid	Deep SVDD	SSAD Raw	SSAD Hybrid	Supervised Classifier	Deep SAD
arrhythmia	84.5±3.9	76.7±6.2	74.6±9.0	**86.7**±4.0	78.3±5.1	39.2±9.5	75.9±8.7
cardio	98.5±0.3	82.8±9.3	84.8±3.6	**98.8**±0.3	86.3±5.8	83.2±9.6	95.0±1.6
satellite	95.1±0.2	68.6±4.8	79.8±4.1	**96.2**±0.3	86.9±2.8	87.2±2.1	91.5±1.1
satimage-2	99.4±0.8	96.7±2.1	98.3±1.4	**99.9**±0.1	96.8±2.1	**99.9**±0.1	**99.9**±0.1
shuttle	99.4±0.9	94.1±9.5	86.3±7.5	**99.6**±0.5	97.7±1.0	95.1±8.0	98.4±0.9
thyroid	98.3±0.9	91.2±4.0	72.0±9.7	97.9±1.9	95.3±3.1	97.8±2.6	**98.6**±0.9

2.3.5 Hypersphere Classification

The above introduced Deep SAD method trains a neural network to concentrate the normal data near some center $\boldsymbol{c}$ and maps anomalous samples to be distant from that center, using an inverse squared norm loss. Below, we present another principled deep semi-supervised one-class classification method, that is based on a cross-entropy classification loss, which we call *Hypersphere Classifier* (HSC).

Let $\phi_\omega : \mathcal{X} \to \mathcal{Z}$ again be a neural network with input space $\mathcal{X} \subseteq \mathbb{R}^D$, output space $\mathcal{Z} \subseteq \mathbb{R}^d$, and network weights ω. Moreover, let $\rho : \mathcal{Z} \to [0, 1]$ be a function that maps an output to a probabilistic score. Given labeled data $(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \dots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y}$ with $\mathcal{Y} = \{\pm 1\}$, the binary cross-entropy loss can be formulated as

$$\min_{\omega} \quad -\frac{1}{m} \sum_{j=1}^{m} \left[\left(\frac{\tilde{y}_j + 1}{2} \right) \log \rho(\phi_\omega(\tilde{\boldsymbol{x}}_j)) + \left(\frac{1 - \tilde{y}_j}{2} \right) \log \Big(1 - \rho(\phi_\omega(\tilde{\boldsymbol{x}}_j)) \Big) \right], \tag{2.18}$$

that is, we identify label $y = +1$ with $\rho = 1$ and $y = -1$ with $\rho = 0$ respectively. For standard binary deep classifiers, ρ is typically modeled as a linear layer followed by the sigmoid activation function, so that the decision region is a half-space $S \subset \mathcal{Z}$. In this case, the preimage $\phi_\omega^{-1}(S)$ of S is not guaranteed to be compact. To impose the preimage to be compact—again aiming to learn a decision region for the normal data $(y = +1)$ that is compact similar to Deep SVDD and Deep SAD—we propose to model $\rho : \mathcal{Z} \to [0, 1]$ as a radial basis function $\rho(\boldsymbol{z}) := \exp\left(-\|\boldsymbol{z}\|^2\right)$. For this choice of ρ, (2.18) becomes

$$\min_{\omega} \quad \frac{1}{m} \sum_{j=1}^{m} \left[\left(\frac{\tilde{y}_j + 1}{2} \right) \|\phi_\omega(\tilde{\boldsymbol{x}}_j)\|^2 - \left(\frac{1 - \tilde{y}_j}{2} \right) \log \Big(1 - \exp(-\|\phi_\omega(\tilde{\boldsymbol{x}}_j)\|^2) \Big) \right]. \tag{2.19}$$

Adding unlabeled data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X}$, again assuming these are fairly clean, we can formulate the following overall semi-supervised anomaly detection objective:

$$\begin{aligned} \min_{\omega} \quad & \frac{1}{n+m} \sum_{i=1}^{n} \|\phi_\omega(\boldsymbol{x}_i)\|^2 \\ & + \frac{\eta}{n+m} \sum_{j=1}^{m} \left[\left(\frac{\tilde{y}_j + 1}{2} \right) \|\phi_\omega(\tilde{\boldsymbol{x}}_j)\|^2 - \left(\frac{1 - \tilde{y}_j}{2} \right) \log \Big(1 - \exp(-\|\phi_\omega(\tilde{\boldsymbol{x}}_j)\|^2) \Big) \right], \end{aligned} \tag{2.20}$$

with hyperparameter $\eta > 0$ that balances the labeled and the unlabeled loss terms.

In comparison to standard binary cross-entropy classification with half-space separation, the HSC objective (2.20) tries to separate two classes (normal vs. anomalous) by concentrating one of the classes (the normal class) into a compact region. If labeled data is absent, (2.20) simplifies to $\frac{1}{n} \sum_i \|\phi_\omega(\boldsymbol{x}_i)\|^2$. This, with added weight decay regularization and incorporating center $\boldsymbol{c}$ as the bias term of ϕ_ω's output layer, corresponds to One-Class Deep SVDD (2.11), which provides another interpretation of the Deep SVDD method. Furthermore, the HSC objective is equivalent to Deep SAD for the positive data $(y = +1)$ and similar in shape (but different in rate) for the negative data $(y = -1)$. We define the HSC anomaly score also as $s(\boldsymbol{x}) := \|\phi_\omega(\boldsymbol{x})\|^2$.

Motivated by robust statistics [210, 244] we also consider replacing ρ with other radial functions where the squared-norm is replaced with a robust alternative. We found that using a pseudo-Huber loss [243, 98], $\rho(\boldsymbol{z}) = \exp(-h(\boldsymbol{z}))$ with $h(\boldsymbol{z}) = \sqrt{\|\boldsymbol{z}\|^2 + 1} - 1$, which interpolates between squared and absolute value penalization, consistently yielded the best detection results in the experimental evaluation on incorporating many auxiliary anomalies presented in the next Section 2.3.6. We include a comparison for other choices of the norm in Appendix A.2. Finally, in view of the general empirical one-class classification objective (2.2) we gave at the beginning of this chapter, we think that further theoretical analyses of this class of radial losses may lead to interesting insights and also motivate the formulation of novel (also possibly new shallow) methods.

2.3.6 Experimental Evaluation on Using Many Auxiliary Anomalies

Many deep learning methods have been proposed that incorporate large amounts of unorganized data that is easily accessible online. In NLP, word embedding models, such as word2vec [370], and language models, such as BERT [135] or GPT-3 [79], which are trained on huge unlabeled text corpora from the web in a self-supervised manner, are the current state of the art and responsible for significant improvements on various NLP tasks. In computer vision, supervised pre-training on large auxiliary datasets [603] such as ImageNet [133], and self-supervised pre-training [105] have been found to be effective. Using such pre-trained models as a starting point is standard in many downstream computer vision tasks.

Hendrycks et al. [221] have introduced the idea of utilizing large unstructured data also for the task of anomaly detection by considering this auxiliary data as anomalous, which they call Outlier Exposure (OE) as mentioned previously. OE makes the assumption that the unstructured data is very unlikely to correspond to what is normal in a given application, but most likely is anomalous in most cases. Although auxiliary anomalies may not be representative for anomalies at testing time (i.e., do not follow $\mathbb{P}^-$), the underlying hypothesis of OE is that this auxiliary data is nevertheless informative for a respective domain (e.g., natural images or the English language in general) and useful to learn an improved representation of the normal data. Exposing a normal model of cat images to random natural images (possibly including images of other animals), for instance, most likely is informative for learning an improved semantic representation of cats.

In the following, we test the above hypothesis and the value of having many auxiliary anomalies available for training for the two deep semi-supervised one-class classification methods, Deep SAD and HSC, we have introduced above.

Setup We consider the CIFAR-10 and ImageNet one vs. rest benchmarks following Hendrycks et al. [222]. That is, in each setup one of the dataset classes is considered normal and the other respective classes are considered anomalous. In every setup, we train a model using only the training set of the respective normal class as well as random samples from a large OE dataset that is disjoint from the ground-truth

anomaly classes at testing time. We use the same auxiliary OE datasets as used in recent literature [221, 222]. For the CIFAR-10 benchmark, which comprises 10 classes, we use the 80 Million Tiny Images dataset [548] as OE (with CIFAR-10 and CIFAR-100 images removed). The ImageNet benchmark contains 30 classes from the ImageNet-1K dataset [133], for which we use the ImageNet-22K dataset as OE (with the ImageNet-1K classes removed). Experiments are iterated over all classes and repeated for multiple random seeds.

Competitors We compare Deep SAD and HSC to recent deep anomaly detection methods that have shown state-of-the-art results on the two benchmarks. These include a self-supervised method based on predicting Geometric Transformations (GT) [181], which subsequently has been improved in [222] (GT+). GT+ has been used with and without OE. We further include the results of a Focal loss classifier [327], which is a binary classifier that specifically addresses class imbalance, that is trained with OE [222]. Finally, we add the results of an autoencoder (AE), and for CIFAR-10 also the results for shallow and Deep SVDD, as unsupervised baselines.

Network Architectures and Optimization We use the same network ϕ_ω in each experimental setup for Deep SAD and HSC. On CIFAR-10, we use a LeNet-type network having three convolutional layers with max-pooling, followed by two fully connected layers. We use (leaky) ReLU activations and apply batch normalization [248] in this network. On ImageNet, we use the same WideResNet [602] as [222], which has ResNet-18 as its architectural backbone. We use Adam [276] for optimization and balance every batch to contain 128 normal and 128 OE samples during training. We apply standard data augmentation using color jitter, random cropping, horizontal flipping, and Gaussian pixel noise.

Results and Discussion The results on the CIFAR-10 and ImageNet one vs. rest benchmarks are shown in Table 2.4 and Table 2.5 respectively. For ImageNet, we report the mean AUC over the 30 classes here and provide the results on all individual classes in Appendix C.2. First, we can observe that using OE results in a markedly improved detection performance on both one vs. rest benchmarks. On CIFAR-10, Deep SAD and HSC achieve a detection performance of 94.5 and 95.9 AUC respectively, whereas unsupervised Deep SVDD resides at 64.8 AUC. GT+ with OE performs similar to Deep SAD and HSC on CIFAR-10. On ImageNet, Deep SAD and HSC show an improved detection performance over GT+. Comparing Deep SAD to HSC, we see that HSC slightly outperforms Deep SAD on both benchmarks. However, using the squared L^2-norm with HSC yields similar results to Deep SAD (see ablation in Appendix A.2), so the advantage of HSC seems to be mainly due to using the robust pseudo-Huber loss, which arguably seems reasonable to use in the OE setting, where there is a lot of variation in the auxiliary OE corpus. Moreover, we remark that the self-supervised methods, GT and GT+, without OE show a marked improvement on the CIFAR-10 benchmark over the other unsupervised methods.

This indicates the potential of self-supervised methods for introducing inductive biases towards learning semantic representations (see also discussion in 5.2.5), which shows advantageous on these object level one vs. rest image benchmarks.

Table 2.4: Detection performance in mean AUC in % (over 10 seeds) for various methods on the CIFAR-10 one vs. rest benchmark using 80 Million Tiny Images as OE. Results taken from the literature are marked with an asterisk [181, 222].

	without OE					with OE			
	SVDD	AE	DSVDD	GT*	GT+*	GT+*	Focal*	DSAD	HSC
airplane	65.6	59.1	61.7	74.7	77.5	90.4	87.6	94.2	**96.3**
automobile	40.9	57.4	65.9	95.7	96.9	**99.3**	93.9	98.1	98.7
bird	65.3	48.9	50.8	78.1	87.3	**93.7**	78.6	89.8	92.7
cat	50.1	58.4	59.1	72.4	80.9	88.1	79.9	87.4	**89.8**
deer	75.2	54.0	60.9	87.8	92.7	**97.4**	81.7	95.0	96.6
dog	51.2	62.2	65.7	87.8	90.2	**94.3**	85.6	93.0	94.2
frog	71.8	51.2	67.7	83.4	90.9	97.1	93.3	96.9	**97.9**
horse	51.2	58.6	67.3	95.5	96.5	**98.8**	87.9	96.8	97.6
ship	67.9	76.8	75.9	93.3	95.2	**98.7**	92.6	97.1	98.2
truck	48.5	67.3	73.1	91.3	93.3	**98.5**	92.1	96.2	97.4
mean	58.8	59.4	64.8	86.0	90.1	95.6	87.3	94.5	**95.9**

Table 2.5: Detection performance in mean AUC in % (over 30 classes and 10 seeds) on the ImageNet-1K one vs. rest benchmark using ImageNet-22K (with the 1K classes removed) as OE. Results taken from the literature are marked with an asterisk [222].

	without OE	with OE			
	AE	Focal*	GT+*	DSAD	HSC
mean	56.0	56.1	85.7	96.7	**97.3**

Lastly, we note that in [471], we present further results showing that using standard BCE classification (and a re-implementation of the Focal loss) with OE, surprisingly yields competitive detection results on the CIFAR-10 and ImageNet one vs. rest image benchmarks as well. Interestingly, the detection performance moreover is already competitive when using relatively few (~128) OE examples. One possible hypothesis for this phenomenon, which we explore in [471], is that the multiscale structure of images makes few example anomalies exceptionally informative. Understanding these counter-intuitive findings presents an interesting question to answer in future work. However, also note that this observation may be limited to object classes considered in the typical one vs. rest benchmarks, where anomalies for testing are fairly structured and distinct (objects from different classes), which is why more diverse and challenging anomaly detection benchmarks are needed in the community (see discussion in Section 5.2.4). In an application to detect more subtle defects in manufacturing in the next Chapter 3, we found the use of OE from general natural images to be limited (see experiments in Section 3.1.2).

Conclusions from this chapter:

- Deep SVDD introduces a deep one-class classification method for unsupervised anomaly detection, which extends the one-class classification approach from fixed features towards learning data representations.
- Deep one-class classification can significantly improve anomaly detection performance over shallow methods on complex data (e.g., images).
- A key challenge in deep one-class classification is avoiding a trivial feature map collapse, which can be addressed through introducing constraints and regularization.
- The Deep SAD and HSC methods present generalizations of Deep SVDD to the semi-supervised anomaly detection setting.
- Including few true anomalies or many auxiliary anomalies can both significantly improve anomaly detection performance.

Parts of this chapter are mainly based on:

[466] L. Ruff*, R. A. Vandermeulen*, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep One-Class Classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4390–4399, 2018.

[469] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, M. Kloft. Deep Semi-Supervised Anomaly Detection. In *International Conference on Learning Representations*, 2020.

With added content from:

[111] P. Chong, L. Ruff, M. Kloft, and A. Binder. Simple and Effective Prevention of Mode Collapse in Deep One-Class Classification. In *International Joint Conference on Neural Networks*, pages 1–9, 2020.

[467] 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 *ICML 2019 Workshop on Uncertainty & Robustness in Deep Learning*, 2019.

[471] L. Ruff, R. A. Vandermeulen, B. J. Franks, K.-R. Müller, and M. Kloft. Rethinking Assumptions in Deep Anomaly Detection. In *ICML 2021 Workshop on Uncertainty & Robustness in Deep Learning*, 2021.

3 Applications to Computer Vision and NLP

The deep one-class classification methods we have introduced in Chapter 2 can be applied to any data type in any domain. In this chapter, we introduce two deep one-class classification variants that are also based on the basic principle of learning a concentrated feature space for the normal data, but additionally integrate domain-specific particularities. Fully Convolutional Data Description incorporates the property of spatial coherence that is important in computer vision, i.e. that neighboring pixels are correlated, by using a fully convolutional network architecture [342, 398], which produces an explanation heatmap in the output. Context Vector Data Description incorporates the aspect of multi-context found in NLP, i.e. that text samples can be interpreted and put into different semantic contexts, by using the self-attention mechanism which also enables model interpretability.

3.1 Explainable One-Class Classification for Images

Deep one-class classification methods learn a transformation that concentrates normal data samples in feature space causing anomalies to be mapped away. Because this transformation can be highly non-linear, finding interpretations poses a significant challenge. In this section, we present an explainable deep one-class classification method for image data, called *Fully Convolutional Data Description* (FCDD), where the mapped samples themselves are also an explanation heatmap. In an experimental evaluation on the standard Fashion-MNIST, CIFAR-10, and ImageNet one vs. rest benchmarks, we find that FCDD yields competitive detection performance while providing transparent explanations. Moreover, on the recent MVTec-AD manufacturing dataset, which offers ground-truth anomaly maps, FCDD achieves a new state of the art in the unsupervised setting. Similar to Deep SAD and HSC (see Section 2.3), FCDD can also take advantage of labeled anomalies in a semi-supervised setting, where it can additionally incorporate ground-truth anomaly maps into training. We find that even using few labeled anomalies (together with the respective anomaly maps) can significantly improve anomaly localization performance.

While there has been much work on deep anomaly detection in the recent years, there is limited work on making such approaches explainable. However, explanations are needed in industrial applications to meet safety and security requirements [19, 55], avoid unfair social biases [204], or support human experts in decision making [253, 377, 482, 58]. One typically makes visual anomalies explainable by annotating pixels with an anomaly score. In some applications, such as localizing tumors in cancer detection [435], these annotations are the primary goal of detection. For autoencoders (see Section 4.2.3), the reconstruction error is usually used as an anomaly score and the pixel-wise error can be naturally used as an anomaly explanation heatmap [54]. Some recent works have also incorporated attention mechanisms into reconstruction models that can be used for explanation [561, 337]. In video anomaly detection, Sabokrou et al. [475] have used a pre-trained fully convolutional architecture together with a sparse autoencoder to extract 2D features and provide bounding boxes for anomaly localization. One drawback of reconstruction methods is that there are (so far) no principled way for incorporating known anomalies during training.

3.1.1 The FCDD Method

FCDD is the utilization of a fully convolutional network in conjunction with Deep SVDD (Section 2.2.1) and HSC (Section 2.3.5), so that the mapped images are themselves an image corresponding to a downsampled anomaly heatmap. The pixels in this heatmap that are far from the center correspond to anomalous regions in the input image. FCDD does this by only using convolutional and pooling layers, thereby limiting the receptive field of each output pixel.

Fully Convolutional Architecture FCDD uses a fully convolutional network (FCN) [342, 398] $\phi_\omega : \mathbb{R}^{c \times h \times w} \to \mathbb{R}^{u \times v}$ that maps an image to a matrix of features, using alternating convolutional and pooling layers only, and does not contain any fully connected layers. In this context, pooling can be seen as a special kind of convolution with fixed parameters.

A defining property of a convolutional layer is that each pixel of its output only depends on a small region of its input, known as the output pixel's receptive field. Since the output of a convolution is produced by moving a filter over the input image, each output pixel has the same relative position as its associated receptive field in the input. For instance, the lower-left corner of the output representation has a corresponding receptive field in the lower-left corner of the input image, etc. (see Figure 3.1). The output of several stacked convolutions also has receptive fields of limited size and consistent relative position, though their size grows with the number of layers. Due to this, FCNs incorporate the assumption of spatial coherence.

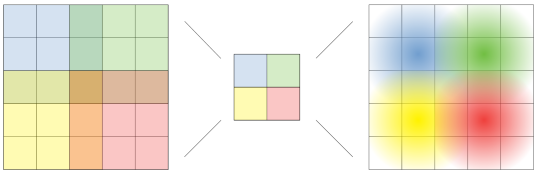


Figure 3.1: Visualization of a 3×3 convolution followed by a 3×3 transposed convolution with a Gaussian kernel, both using a stride of 2.

FCDD Objective Let $\phi_\omega : \mathbb{R}^{c\times h\times w} \to \mathbb{R}^{u\times v}$ be a FCN with network weights ω. Moreover, define $A_\omega(\boldsymbol{x}) := (\sqrt{\phi_\omega(\boldsymbol{x})^2 + 1} - 1)$, that is $A_\omega(\boldsymbol{x})$ is the pseudo-Huber loss (see also Section 2.3.5) applied to the FCN output matrix $\phi_\omega(\boldsymbol{x})$, where all operations are applied element-wise. Given n unlabeled images $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ with $\mathcal{X} \subseteq \mathbb{R}^{c\times h\times w}$ and m labeled images $(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \ldots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y}$ with $\mathcal{Y} = \{\pm 1\}$, where again $\tilde{y} = +1$ denotes known normal images and $\tilde{y} = -1$ known anomalies, respectively, we define the FCDD objective as:

$$\begin{aligned}\min_{\omega} \quad & \frac{1}{n+m}\sum_{i=1}^{n}\frac{\|A_\omega(\boldsymbol{x}_i)\|_1}{uv} \\ & + \frac{\eta}{n+m}\sum_{j=1}^{m}\left(\frac{\tilde{y}_j+1}{2}\right)\frac{\|A_\omega(\tilde{\boldsymbol{x}}_j)\|_1}{uv} - \left(\frac{1-\tilde{y}_j}{2}\right)\log\left(1-\exp\left(-\frac{\|A_\omega(\tilde{\boldsymbol{x}}_j)\|_1}{uv}\right)\right),\end{aligned} \tag{3.1}$$

where $\|A_\omega(\boldsymbol{x})\|_1$ is the sum of all entries in $A_\omega(\boldsymbol{x})$, which are all positive, and hyperparameter $\eta > 0$ again controls the balance between the labeled and the unlabeled term (see also (2.13) in Section 2.3.2).

We omit the usual center $\boldsymbol{c}$ in the FCDD objective (3.1), since we always train FCDD using (true, auxiliary, or artificial) negative examples, which prevents a hypersphere collapse (see Sections 2.2.2 and 2.2.3). In our implementation, we include and optimize a bias term in the last layer of our networks that corresponds to $\boldsymbol{c}$. As discussed in Section 2.3.1, labeled anomalous samples can be a collection of auxiliary images which are different from the collection of normal images (Outlier Exposure), for example one of the many large collections of images that are freely available like 80 Million Tiny Images [548] or ImageNet [133]. When one has access to ground-truth anomalies, that is, images that are representative of anomalies that will be seen at testing time, we find that even using a few examples as the corpus of labeled anomalies performs well. Furthermore, in the absence of *any* sort of known anomalies, one can artificially generate synthetic anomalies, which we find to be very effective for anomaly localization as well.

Objective (3.1) maximizes $\|A_\omega(\boldsymbol{x})\|_1$ for anomalies and minimizes it for normal instances, thus we use $\|A_\omega(\boldsymbol{x})\|_1$ as the anomaly score. Entries of $A_\omega(\boldsymbol{x})$ that contribute to $\|A_\omega(\boldsymbol{x})\|_1$ correspond to regions of the input image $\boldsymbol{x}$ that add to the anomaly score. The shape of these regions depends on the receptive field of the FCN. We include a sensitivity analysis on the size of the receptive field in Appendix A.3, where we find that detection performance is not much affected within a reasonable range of sizes. Note that $A_\omega(\boldsymbol{x})$ has spatial dimensions $u \times v$ and is smaller than the original image dimensions $h \times w$. One could use $A_\omega(\boldsymbol{x})$ directly as a low-resolution heatmap of the image, however it is often desirable to have full-resolution heatmaps. Because we usually lack ground-truth anomaly maps in an anomaly detection setting during training, it is not possible to train an FCN in a supervised way to upsample the low-resolution heatmap $A_\omega(\boldsymbol{x})$ (e.g., as in [398]). For this reason we introduce an upsampling methodology based on the properties of receptive fields.

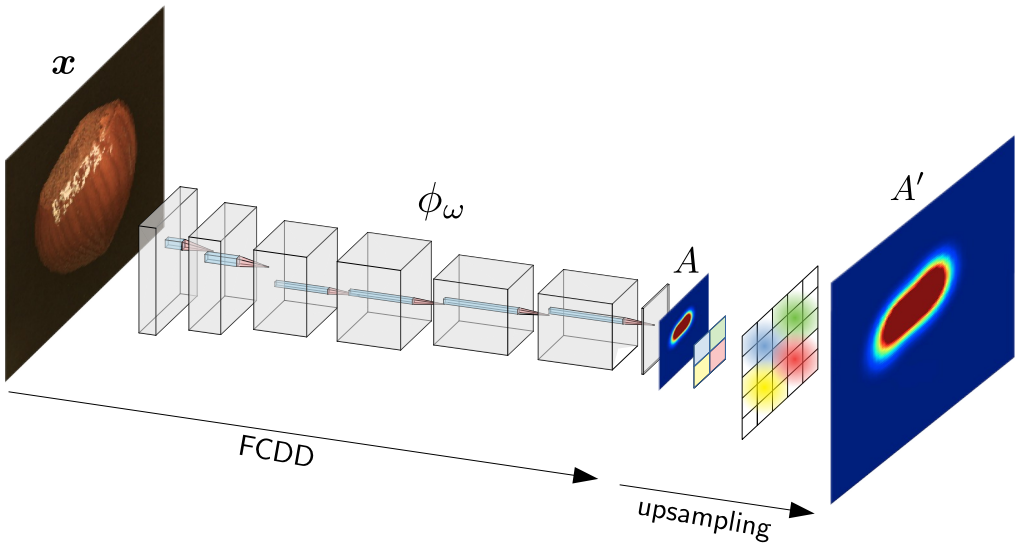


Figure 3.2: FCDD uses a fully convolutional network ϕ_ω with a deep one-class classification objective to produce an anomaly heatmap A of an input $\boldsymbol{x}$. The lower-resolution heatmap A can be upsampled to a full-resolution anomaly heatmap A' via a transposed Gaussian convolution.

Heatmap Upsampling Since we generally do not have access to ground-truth pixel annotations in anomaly detection during training, we cannot learn a deconvolutional type of structure for upsampling. Instead, we suggest a principled way to upsample the lower resolution anomaly heatmap. For every output pixel in $A_\omega(\boldsymbol{x})$ there is a unique input pixel which lies at the center of its receptive field. It has been observed before that the influence of the receptive field for an output pixel decays in a Gaussian manner as one moves away from the center of the receptive field [345]. We use this fact to upsample $A_\omega(\boldsymbol{x})$ by using a strided transposed convolution with a fixed Gaussian kernel (see Figure 3.1 right side). This operation and procedure is described in Algorithm 1, which simply corresponds to a strided transposed convolution. The kernel size is set to the receptive field range of FCDD and the stride to the cumulative stride of FCDD. The variance of the Gaussian kernel can be picked empirically (see Appendix A.4 for details). In Figure 3.2, we give a complete overview of the FCDD method and the process of generating full-resolution anomaly heatmaps.

Algorithm 1 Gaussian Receptive Field Upsampling

Input: $A \in \mathbb{R}^{u \times v}$ (low-res anomaly heatmap)

Output: $A' \in \mathbb{R}^{h \times w}$ (full-res anomaly heatmap)

Define: $[G_2(\mu, \sigma)]_{x,y} := \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-\mu_1)^2+(y-\mu_2)^2}{2\sigma^2}\right)$

$A' \leftarrow 0$
for all output pixels a in A **do**
  $f \leftarrow$ receptive field of a
  $c \leftarrow$ center of field f
  $A' \leftarrow A' + a \cdot G_2(c, \sigma)$
end for
return A'

3.1.2 Experimental Evaluation

In this section, we experimentally evaluate the performance of FCDD both quantitatively and qualitatively. For a quantitative evaluation, we again use the standard AUC measure. For a qualitative evaluation, we compare the heatmaps produced by FCDD to existing deep anomaly detection explanation baselines. As baselines, we consider gradient-based methods [515] applied to hypersphere classifier (HSC) models that use unrestricted network architectures (i.e., networks that also have fully connected layers) as well as autoencoders [54], where we directly use the pixel-wise reconstruction error as an explanation heatmap. We slightly blur the heatmaps of the baselines with the same Gaussian kernel we use for FCDD, which we found to result in less noisy, more interpretable heatmaps (non-blurred heatmaps are given in the Appendix of [339]).[1]

Standard Anomaly Detection Benchmarks

We first evaluate FCDD on the Fashion-MNIST [589], CIFAR-10 [290], and ImageNet [133] one vs. rest benchmarks [466, 130, 181, 13, 1, 424, 574, 222, 52], where again always one class is considered as the normal class and the rest of the classes are used as anomalies at testing time. The purpose of this experiment is to see the impact of using a fully convolutional architecture on detection performance, which is more restricted than a general architecture, but which we use in FCDD to obtain anomaly explanations. For training, we only use normal data samples as well as random samples from some auxiliary Outlier Exposure (OE) [221] dataset, which is separate from the ground-truth anomaly classes again following Hendrycks et al. [221, 222]. We report the mean AUC over all classes for each dataset.

Fashion-MNIST We consider each of the ten Fashion-MNIST classes in a one vs. rest setup. We train Fashion-MNIST using EMNIST [116] or grayscaled CIFAR-100 [290] as OE, where we found the latter to slightly outperforms the former ($\sim$3 AUC percent points). On Fashion-MNIST, we use a network that consists of three convolutional layers with batch normalization, separated by two pooling layers.

CIFAR-10 We consider each of the ten CIFAR-10 classes in a one vs. rest setup. As OE, we use CIFAR-100, which does not share any classes with CIFAR-10. We use a model similar to LeNet-5 [303], but decrease the kernel size to three, add batch normalization, and replace the fully connected layers and last max-pool layer with two further convolutions.

ImageNet We consider 30 classes from ImageNet1k for the one vs. rest setup following [222] as above. As OE, we use ImageNet22k with ImageNet1k classes

[1] A `PyTorch` implementation of FCDD and code to reproduce the results is available at: `https://github.com/liznerski/fcdd`

removed [222]. We use an adaptation of VGG11 [514] with batch normalization, suitable for inputs resized to 224×224 (see Appendix B.2 for architecture details).

Competitors We report the results from recent state-of-the-art deep anomaly detection methods. Methods that do not incorporate OE are the autoencoder (AE), Deep SVDD, self-supervised anomaly detection based on Geometric Transformations (GT) [181], and an improved variant of GT by Hendrycks et al. [222] (GT+). Methods that use OE are the Focal loss classifier [222], also GT+, Deep SAD, and HSC.

Table 3.1: Mean AUC (over all classes and 5 seeds per class) for Fashion-MNIST, CIFAR-10, and ImageNet. Results taken from the literature are marked with an asterisk [52, 181, 222].

	without OE				with OE				
	AE	DSVDD	GT*	GT+*	GT+*	Focal*	DSAD	HSC	FCDD
F-MNIST	0.82	0.93	**0.94**	×	×	×	×	×	0.89
CIFAR-10	0.59	0.65	0.86	0.90	**0.96**	0.87	0.95	**0.96**	0.95
ImageNet	0.56	×	×	×	0.86	0.56	**0.97**	**0.97**	0.94

Quantitative Results The mean AUC detection performance on the three benchmarks are reported in Table 3.1. We can see that FCDD, despite using a restricted FCN architecture to improve explainability, achieves a performance that is close to state-of-the-art methods. Note also that the autoencoder yields a detection performance that is close to random guessing on the more complex CIFAR-10 and ImageNet datasets, which puts the use of a reconstruction error for semantic detection tasks into question. Another reason for this certainly is that the autoencoder does not take advantage of OE. We provide the individual results for all classes in Appendix C.3.

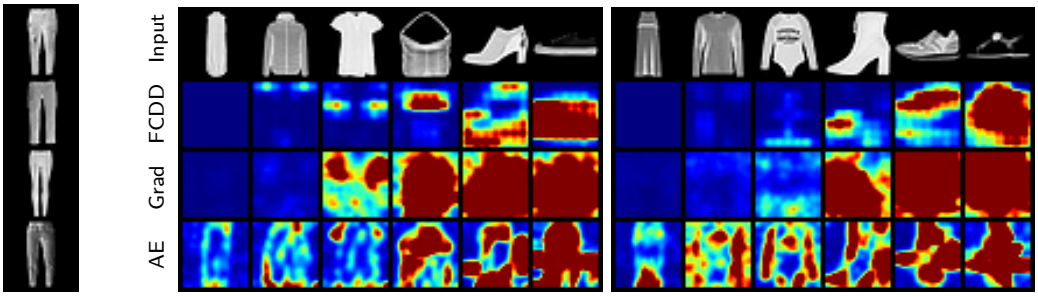


Figure 3.3: Anomaly heatmaps for anomalous test images of a Fashion-MNIST model trained on normal class `trousers` (normal images are shown in (a)). In (b) CIFAR-100 and in (c) EMNIST was used for OE, respectively. Columns are ordered by increasing anomaly score from left to right, i.e. what is found to be the most normal looking anomaly on the left to the most anomalous looking anomaly on the right.

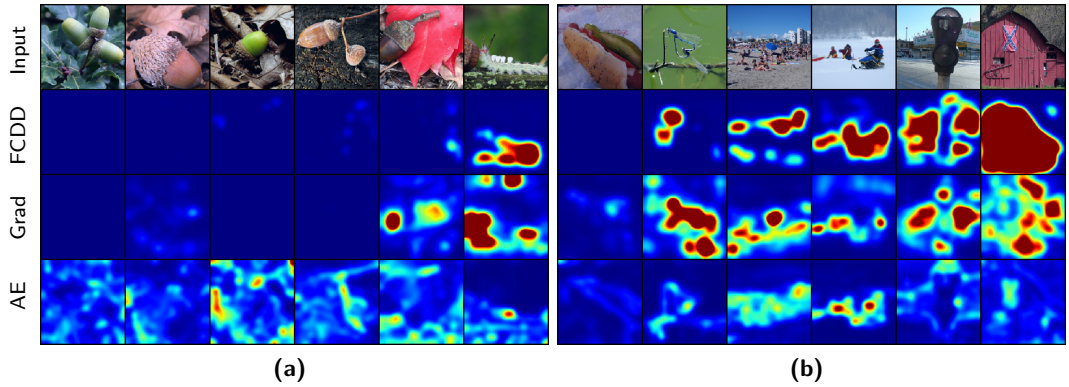


Figure 3.4: Anomaly heatmaps of an ImageNet model trained on normal class `acorns`. Here (a) are normal samples and (b) are anomalous samples. Columns are ordered by increasing anomaly score from left to right, i.e. what is found to be the most normal looking on the left to the most anomalous looking on the right for (a) normal samples and (b) anomalies.

Qualitative Results In Figures 3.3 and 3.4, we show some heatmaps for Fashion-MNIST and ImageNet respectively. For a Fashion-MNIST model trained on the normal class `trousers`, the heatmaps show that FCDD correctly highlights horizontal elements as being anomalous, which seems reasonable since trousers are vertically aligned. Qualitatively, we do not see systematic differences in the Fashion-MNIST heatmaps between using CIFAR-100 or EMNIST as OE. For an ImageNet model trained on the normal class `acorns`, we observe that colors seem to be fairly relevant features with green and brown areas tending to be seen as more normal, and other colors being deemed anomalous, for example the red barn or the white snow. Nonetheless, FCDD also seems capable of using more semantic features, for example it identifies the green caterpillar as being anomalous and recognizes the acorn in front of the red leaf as being normal despite the red background.

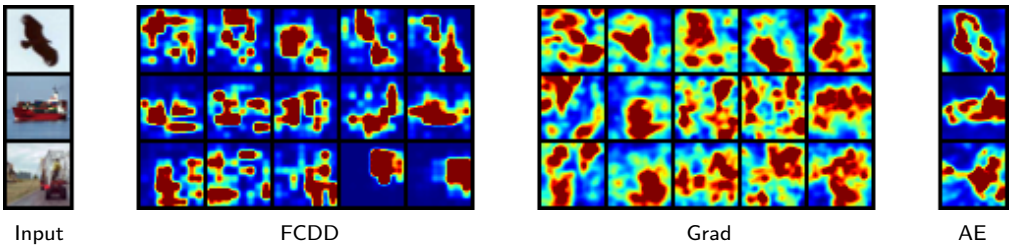


Figure 3.5: Anomaly heatmaps for three anomalous test samples on a CIFAR-10 model trained on normal class `airplane`. The second, third, and fourth blocks show the heatmaps of FCDD, gradient-based heatmaps of HSC, and AE heatmaps respectively. For FCDD and Grad, we grow the number of OE samples from 2, 8, 128, 2048 to full OE. The AE does not incorporate OE.

To see the qualitative impact of using varying amounts of OE on FCDD heatmaps, we run an experiment on CIFAR-10 while increasing the number of OE samples.

Figure 3.5 shows the heatmaps for CIFAR-10 models trained on the normal class `airplane`. We can see that, as the number of OE samples grows, the FCDD explanation heatmaps tend to concentrate more on the primary object in an image (i.e., the bird, ship, and truck), as opposed to highlighting parts of the background. This is another indication that OE seems to help with learning more semantic features.

Baseline Explanations We found the gradient-based heatmaps to often result in centered blobs which lack spatial information (see Figure 3.5) and thus not useful for explaining anomalies. The AE heatmaps, being directly tied to the reconstruction error anomaly score, look more reasonable. Note again, however, that it is not straightforward how to include auxiliary OE samples or labeled anomalies into an AE approach, which leaves them with a poorer semantic detection performance (see Table 3.1). Overall we find the proposed FCDD anomaly heatmaps to yield good and consistent visual interpretations.

Explaining Defects in Manufacturing

Here we compare the performance of FCDD on the MVTec-AD dataset of defects in manufacturing [54]. This datasets offers annotated ground-truth anomaly segmentation maps for testing, thus allowing a quantitative evaluation of model explanations. MVTec-AD contains five texture and ten object classes of high-resolution RGB images with up to 1024×1024 pixels, where anomalous test samples are further categorized in up to eight defect types, depending on the class. We follow Bergmann et al. [54] and compute the AUC from the pixel-wise explanation heatmap scores, using the given (binary) anomaly segmentation maps as ground-truth pixel labels. We then report the mean over all samples of this "explanation AUC" for a quantitative evaluation of explanation performance. For FCDD, we use a network that is based on a VGG11 network pre-trained on ImageNet, where we freeze the first ten layers, followed by additional fully convolutional layers that we optimize.

Synthetic Anomalies We found OE with a natural image dataset like ImageNet not particularly informative for MVTec-AD, since anomalies here are subtle defects of the normal class, rather than being semantically out of class (see defects shown in Figure 3.7). For this reason, we generate synthetic anomalies using a sort of "confetti noise," a simple noise model that inserts colored blobs into normal images to reflect the local nature of the defects. Figure 3.6 shows examples of synthetic anomalies generated in this way.

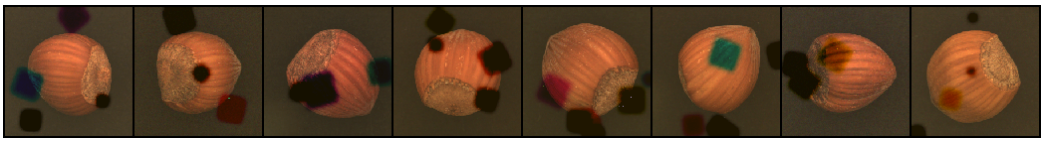


Figure 3.6: Synthetic anomalies generated on MVTec-AD `hazelnuts` using the confetti noise.

Semi-Supervised FCDD One advantage of FCDD in comparison to reconstruction methods, is that it can readily incorporate labeled anomalies along with their corresponding ground-truth anomaly maps. To take advantage of ground-truth heatmaps, we can simply train the FCDD objective on a pixel level, taking the upsampled output $A'_\omega(\boldsymbol{x})$ instead of the lower dimensional $A_\omega(\boldsymbol{x})$ in the objective (3.1). To see the effect of having even only a few labeled anomalies available for training, we pick for each MVTec-AD class just *one* true anomalous sample per defect type at random and add it to the training set. This results in only 3–8 anomalous training samples.

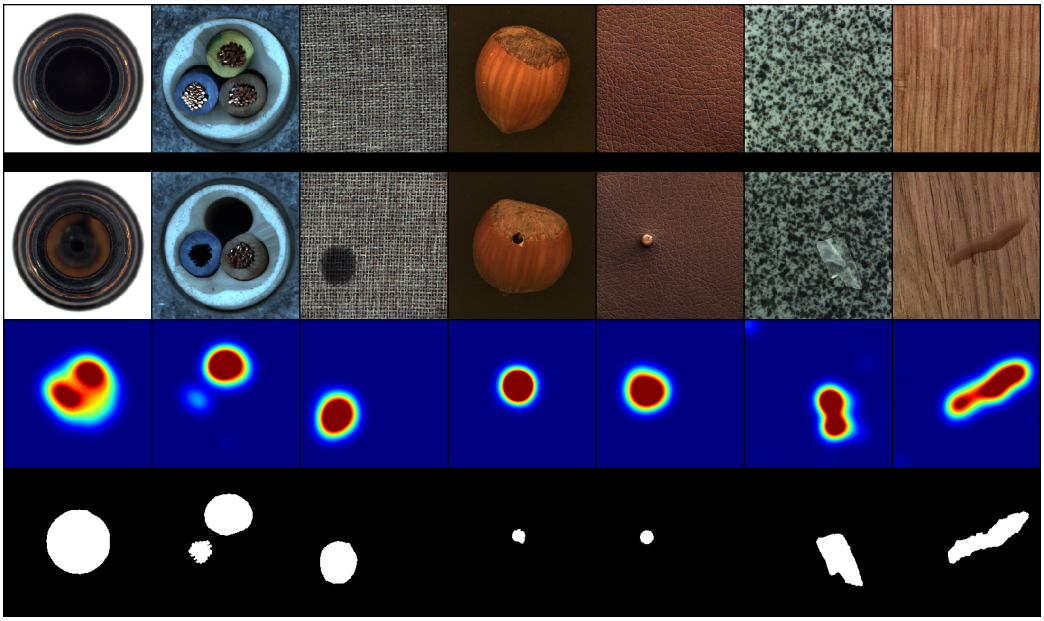


Figure 3.7: FCDD explanation heatmaps on MVTec-AD [54]. The rows from top to bottom show: (1) normal samples; (2) anomalies involving defects (from left to right: contaminated bottle (top view); cable missing a wire; carpet with color stain; hazelnut with hole; leather with glue drop; tile with glue strip; wood with liquid); (3) FCDD anomaly heatmaps; (4) ground-truth anomaly maps.

Results and Discussion In Figure 3.7, we show some explanation heatmaps of FCDD trained on various classes of MVTec-AD. We can see that the FCDD defect explanation heatmaps match the ground-truth anomaly maps well. One interesting observation is that the heatmaps for the hole and glue drop defects on the hazelnut and leather patch, respectively, cover larger regions than the actual defect, which likely is an effect of learning a lower-resolution heatmap and using Gaussian upsampling with FCDD. In Table 3.2, we provide the quantitative results to get a systematic overview of explanation performance. We can see that FCDD improves the anomaly localization performance over previous methods with a new state of the art of 0.92 pixel-wise mean AUC in the unsupervised setting. In the semi-supervised setting, using only one anomalous sample with corresponding anomaly map per defect class,

the explanation performance can be further improved to 0.96 pixel-wise mean AUC. This again shows the potential and value of using few labeled anomalies in a semi-supervised approach to anomaly detection (see also Section 2.3.4). Finally note that FCDD also shows the most consistent results across classes.

Table 3.2: Pixel-wise mean AUC scores for all classes of the MVTec-AD dataset [54]. For comparison, we include the baselines presented in the original MVTec-AD paper and previous works that include the MVTec-AD benchmark. The methods are Self-Similarity and L2 Autoencoder [54], AnoGAN [488, 54], CNN Feature Dictionaries [389, 54], Visually Explained Variational Autoencoder [337], Superpixel Masking and Inpainting [322], Gradient Descent Reconstruction with VAEs [132], and Encoding Structure-Texture Relation with P-Net for AD [619].

		unsupervised									semi-sup.
		*AE-SS**	*AE-L2**	*AnoGAN**	*CNNFD**	*VEVAE**	*SMAI**	*GDR**	*P-NET**	*FCDD*	*FCDD*
Textures	carpet	0.87	0.59	0.54	0.72	0.78	0.88	0.74	0.57	**0.96**	**0.99**
	grid	0.94	0.9	0.58	0.59	0.73	0.97	0.96	**0.98**	0.91	0.95
	leather	0.78	0.75	0.64	0.87	0.95	0.86	0.93	0.89	**0.98**	**0.99**
	tile	0.59	0.51	0.5	0.93	0.80	0.62	0.65	**0.97**	0.91	**0.98**
	wood	0.73	0.73	0.62	0.91	0.77	0.80	0.84	**0.98**	0.88	0.94
Objects	bottle	0.93	0.86	0.86	0.78	0.87	0.86	0.92	**0.99**	0.97	0.96
	cable	0.82	0.86	0.78	0.79	0.90	**0.92**	0.91	0.70	0.90	**0.93**
	capsule	**0.94**	0.88	0.84	0.84	0.74	0.93	0.92	0.84	0.93	**0.95**
	hazelnut	0.97	0.95	0.87	0.72	**0.98**	0.97	**0.98**	0.97	0.95	0.97
	metal nut	0.89	0.86	0.76	0.82	**0.94**	0.92	0.91	0.79	**0.94**	**0.98**
	pill	0.91	0.85	0.87	0.68	0.83	0.92	**0.93**	0.91	0.81	**0.97**
	screw	0.96	0.96	0.8	0.87	0.97	0.96	0.95	**1.00**	0.86	0.93
	toothbrush	0.92	0.93	0.90	0.77	0.94	0.96	**0.99**	**0.99**	0.94	0.95
	transistor	0.90	0.86	0.80	0.66	**0.93**	0.85	0.92	0.82	0.88	0.90
	zipper	0.88	0.77	0.78	0.76	0.78	0.90	0.87	0.90	**0.92**	**0.98**
	mean	0.86	0.82	0.74	0.78	0.86	0.89	0.89	0.89	**0.92**	**0.96**
	std. dev.	0.10	0.13	0.13	0.10	0.09	0.09	0.09	0.12	**0.04**	**0.02**

3.2 Multi-Context One-Class Classification for Text

Detecting anomalies on text data can be useful for many applications such as discerning anomalous web content (e.g., posts, reviews, or product descriptions), automated content management, spam detection, or characterizing news articles so as to identify similar or novel topics. However, there exist only few anomaly detection methods that are specific to text. Research on learning representations of text (and explaining the learned representations [31]) has seen great advances recently, which has been key to the progress on numerous downstream NLP tasks using transfer learning. Given the exceptional impact that universal vector embeddings of words [49, 118] such as word2vec [370], GloVe [422], or fastText [67, 261], and dynamic vector embeddings of text via language models such as ELMo [425], BERT [135], or GPT-3 [79] have had on NLP, it is somewhat surprising that there has been little work on adapting anomaly detection techniques to utilize such unsupervised pre-trained models.

Previous anomaly detection methods for text still largely rely on bag-of-words (BoW) text representations. Manevitz and Yousef [356] have studied one-class classification of documents using the OC-SVM [496] and a simple autoencoder [357]. Liu et al. [333] have considered a partially supervised classification approach for documents that is similar to one-class classification. Kannan et al. [265] have introduced a nonnegative matrix factorization method for anomaly detection on text that is optimized via block coordinate descent. Mahapatra et al. [351] have proposed a LDA clustering variant that incorporates external contextual information for detecting anomalies. All these works, however, only consider text representations based on document-to-word co-occurrence statistics. Other text-specific approaches that have been proposed rely on specific hand-crafted features for their particular domains or relevant types of anomalies [206, 294].

In this section, we introduce a one-class classification method, Context Vector Data Description (CVDD), that takes advantage of word embedding models for anomaly detection on text. Starting with pre-trained word embeddings, CVDD finds a collection of transforms to map variable-length sequences of word embeddings to a collection of fixed-length text representations via a multi-head self-attention mechanism. These representations are trained along with a collection of *context vectors* such that the representations and context vectors are closely aligned while keeping the context vectors diverse. Modeling multiple contexts enables CVDD to capture multiple themes and concepts of normalcy in an unlabeled text corpus, which may, for example, correspond to distinct yet non-anomalous topics. These contexts allow us to perform contextual anomaly detection and, together with the self-attention weights, make a trained CVDD model also interpretable. We demonstrate the effectiveness of CVDD quantitatively as well as qualitatively on the well-known Reuters, 20 Newsgroups, and IMDB Movie Reviews datasets.

Since publication [468], CVDD has also been extended to time series anomaly

detection [506]. Moreover, new anomaly detection methods that utilize word embeddings have been developed [164, 599].

3.2.1 The CVDD Method

In the following, we introduce CVDD, a self-attentive, multi-context one-class classification method for unsupervised anomaly detection on text. We first describe the multi-head self-attention mechanism we use, then introduce the CVDD objective, and afterwards discuss regularization, optimization, and further properties of CVDD.

Multi-Head Self-Attention

Let $S = (\boldsymbol{w}_1, \dots, \boldsymbol{w}_l) \in \mathbb{R}^{D \times l}$ be a sentence or, more generally, a sequence of $l \in \mathbb{N}$ words (e.g., phrase or document), where each word is represented by some D-dimensional vector (usually a one-hot vector indexing a word in a dictionary). Given some pre-trained word embedding model, let $H = (\boldsymbol{h}_1, \dots, \boldsymbol{h}_l) \in \mathbb{R}^{d \times l}$ be the corresponding d-dimensional vector embeddings of the words in S. The vector embedding H might be some universal word embedding (e.g., GloVe, fastText) or the hidden vector activations of sentence S given by some language model (e.g., ELMo, BERT).

The aim of *multi-head self-attention* [328] is to define a transformation that accepts sentences $S^{(1)}, \dots, S^{(n)}$ of varying lengths $l^{(1)}, \dots, l^{(n)}$ and returns vectors of fixed length, thereby allowing us to apply more standard ML techniques. The idea here is to find such a fixed-length vector representation of size d via a convex combination of the word embeddings H of a sentence S. The coefficients of this convex combination are adaptive weights that are learned during training.

We now describe the model in more detail. Given the word embeddings $H \in \mathbb{R}^{d \times l}$ of a sentence S, the first step of the self-attention mechanism is to compute an attention matrix $A \in (0, 1)^{l \times r}$ via

$$A = \operatorname{softmax}\Big(\tanh(H^\top W_1) W_2\Big), \tag{3.2}$$

with weights $W_1 \in \mathbb{R}^{d \times d_a}$ and $W_2 \in \mathbb{R}^{d_a \times r}$. The tanh-activation is applied element-wise and the softmax column-wise, thus making each vector $\boldsymbol{a}_k$ of the attention matrix $A = (\boldsymbol{a}_1, \dots, \boldsymbol{a}_r)$ a positive vector that sums to one, i.e. a weighting vector. The r vectors $\boldsymbol{a}_1 \dots, \boldsymbol{a}_r$ are called *attention heads* where each head provides a weighting over the words in the sentence. The dimension d_a specifies the internal dimensionality and thus sets the complexity of the self-attention module. We now obtain a fixed-length sentence embedding matrix $M = (\boldsymbol{m}_1, \dots, \boldsymbol{m}_r) \in \mathbb{R}^{d \times r}$ from the word embeddings H by applying the self-attention weights A as

$$M = HA. \tag{3.3}$$

That is, each column $\boldsymbol{m}_k \in \mathbb{R}^d$ is a convex combination of the vector embeddings $\boldsymbol{h}_1, \dots, \boldsymbol{h}_l \in \mathbb{R}^d$ with weights $\boldsymbol{a}_k \in \mathbb{R}^l$ given by the respective kth attention head,

i.e. $\boldsymbol{m}_k = H\boldsymbol{a}_k$. Often, a regularization term $\mathcal{R}$ such as

$$\mathcal{R} = \frac{1}{n}\sum_{i=1}^{n} \left\|(A^{(i)\top} A^{(i)} - I)\right\|_{\mathrm{F}}^2 \tag{3.4}$$

is added to a learning objective that uses attention to promote the attention heads to be nearly orthogonal and thus capture distinct views that focus on different aspects of the data. Here, I denotes the $r \times r$ identity matrix, $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm, and $A^{(i)} := A(H^{(i)}; W_1, W_2)$ is the attention matrix corresponding to sample $S^{(i)}$.

The CVDD Objective

To define the CVDD objective, which utilizes the multi-head self-attention mechanism described above to learn distinct contexts (one context per head), we first set a notion of similarity. Let $\mathrm{sim}(u, v)$ be the cosine similarity between two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, i.e.

$$\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \frac{\langle \boldsymbol{u}, \boldsymbol{v} \rangle}{\|\boldsymbol{u}\|\,\|\boldsymbol{v}\|} \in [-1, 1] \tag{3.5}$$

and by $d(\boldsymbol{u}, \boldsymbol{v})$ denote the cosine distance between $\boldsymbol{u}$ and $\boldsymbol{v}$, i.e.

$$d(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{2}\left(1 - \mathrm{sim}(\boldsymbol{u}, \boldsymbol{v})\right) \in [0, 1]. \tag{3.6}$$

As before, let r be the number of attention heads. We now define the context matrix $C = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r) \in \mathbb{R}^{d \times r}$ to be a matrix whose columns $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r$ are vectors in the word embedding space $\mathbb{R}^d$. Given an unlabeled training corpus $S^{(1)}, \ldots, S^{(n)}$ of sentences (or phrases, documents, etc.), which may have different lengths $l^{(1)}, \ldots, l^{(n)}$, and their corresponding word vector embeddings $H^{(1)}, \ldots, H^{(n)}$, we formulate the CVDD objective as:

$$\min_{C, W_1, W_2} \quad \underbrace{\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{r} \sigma_k(H^{(i)})\, d(\boldsymbol{c}_k, \boldsymbol{m}_k^{(i)})}_{=: J_n(C, W_1, W_2)}. \tag{3.7}$$

Here, $\sigma_1(H), \ldots, \sigma_r(H)$ are input-dependent weights, i.e. $\sum_k \sigma_k(H) = 1$, which we specify further below. CVDD thus optimizes a set of vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r \in \mathbb{R}^d$ to be closely aligned with the respective attention-weighted sentence embeddings $\boldsymbol{m}_1^{(i)}, \ldots, \boldsymbol{m}_r^{(i)} \in \mathbb{R}^d$. This leads the network to learn attention weights that extract the most common terms and themes from the data. We call $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r \in \mathbb{R}^d$ *context vectors* as they represent a compact description of the different contexts inherent to the data. For a text sample $S^{(i)}$, the corresponding embedding $\boldsymbol{m}_k^{(i)}$ provides a representation of the text with respect to the kth context. To ensure that we extract different contexts from the data, we add a regularization term to objective (3.7).

Multi-Context Regularization To promote the context vectors $C = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r)$ to capture diverse terms and themes, we regularize them towards orthogonality:

$$\mathcal{R}(C) = \|C^\top C - I\|_{\mathrm{F}}^2. \tag{3.8}$$

Hence, the overall CVDD objective becomes:

$$\min_{C, W_1, W_2} \quad J_n(C, W_1, W_2) + \lambda \mathcal{R}(C), \tag{3.9}$$

where $J_n(C, W_1, W_2)$ is the objective function from (3.7) and $\lambda > 0$ is the regularization hyperparameter. Because CVDD minimizes the cosine distances

$$d(\boldsymbol{c}_k, \boldsymbol{m}_k) = \frac{1}{2}\left(1 - \left\langle \frac{\boldsymbol{c}_k}{\|\boldsymbol{c}_k\|}, \frac{H\boldsymbol{a}_k}{\|H\boldsymbol{a}_k\|} \right\rangle\right), \tag{3.10}$$

regularizing the context vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$ to be orthogonal implicitly regularizes the attention weight vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r$ to be orthogonal as well, which we have also observed empirically. We found that regularizing the context vectors as in (3.8), however, allows for faster, more stable optimization in comparison to regularizing the attention weights as in (3.4). This is likely because in (3.4) $\mathcal{R} = \mathcal{R}_n(W_1, W_2)$ depends on the data and the attention network weights W_1 and W_2 in a non-linear fashion. In comparison, the gradients of $\mathcal{R}(C)$ in (3.8) can be directly computed. Empirically we found that selecting $\lambda \in \{1, 10\}$ yielded reliable results with the desired effect that CVDD learns multiple distinct contexts.

Optimization We optimize the CVDD objective jointly over the self-attention network weights $\{W_1, W_2\}$ and the context vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r$ using SGD and its variants (e.g., Adam [276]). Since the self-attention module is just a two-layer feedforward network, the computational complexity of CVDD is low. However, evaluating a pre-trained model for obtaining the word embeddings may add to the computational cost (e.g., in case of large pre-trained language models) in which case parallelization strategies should be exploited (e.g., using multiple GPUs). We initialize the context vectors with the centroids resulting from running *k-means++* [32] on simple mean sentence embeddings obtained from averaging the word embeddings of a sentence. Empirically, we found this initialization strategy to improve optimization speed and performance.

Weighting Contexts For text data samples such as sentences or a documents, there is a natural motivation to consider multiple embeddings per sample for representation (or contextual representations), because text can often be viewed in multiple contexts, e.g. movie genre, cinematic language, or sentiment for movie reviews. This raises the question of how such multiple, contextual embeddings should be weighted in learning a characterization of a text corpus. For this, we propose to use a parameterized softmax over the r cosine distances of a sample S with embedding H in the CVDD objective (3.7):

$$\sigma_k(H) = \frac{\exp(-\alpha\, d(\boldsymbol{c}_k, \boldsymbol{m}_k(H)))}{\sum_{j=1}^{r} \exp(-\alpha\, d(\boldsymbol{c}_j, \boldsymbol{m}_j(H)))}, \tag{3.11}$$

for $k = 1, \ldots, r$ with $\alpha \geq 0$. The temperature hyperparameter α allows to balance two extreme weighting cases: (i) $\alpha = 0$ which results in all contexts being weighted equally, i.e. $\sigma_k(H) = 1/r$ for all k, and (ii) $\alpha \to \infty$ in which case the softmax approximates the argmin function, i.e. only the closest context $k_{\text{min}} = \text{argmin}_k \, d(\boldsymbol{c}_k, \boldsymbol{m}_k)$ has weight $\sigma_{k_{\text{min}}} = 1$ whereas $\sigma_k = 0$ for $k \neq k_{\text{min}}$ otherwise.

Traditional clustering methods typically only consider the argmin, that is, the closest representatives (e.g., nearest centroid for k-means). For learning multiple sentence embeddings and contexts from data, however, this may be ineffective and result in a poor data representation. This is because optimization may get stuck early in the local minimum of the closest context vectors, which strongly depends on the initialization. Not considering the distances to other context vectors also prevents the extraction of multiple contexts per sample. For this reason, we initially set $\alpha = 0$ in training and then gradually increase the α parameter using some annealing strategy. This way, learning is initially focused on extracting multiple contexts from the data ("exploration") before the embeddings subsequently get fine-tuned w.r.t. their closest contexts ("exploitation").

Contextual Anomaly Score

The CVDD formulation enables us to score the "anomalousness" of a text sample w.r.t. the multiple context vectors, that is, to determine anomalies contextually. We define the anomaly score w.r.t. context k for some sample S with embedding H as

$$s_k(H) = d(\boldsymbol{c}_k, \boldsymbol{m}_k(H)), \tag{3.12}$$

that is, as the cosine distance of the contextual embedding $\boldsymbol{m}_k(H)$ to the respective context vector $\boldsymbol{c}_k$. One straightforward choice for an overall anomaly score then is to take the mean over all contextual anomaly scores:

$$s(H) = \frac{1}{r} \sum_{k=1}^{r} s_k(H). \tag{3.13}$$

Depending on the situation, one might also consider different aggregations of the contextual anomaly scores, however, as different contexts may be more or less relevant in certain applications. Another natural choice would be to consider the minimum over the contextual anomaly scores, $s(H) = \min_k s_k(H)$, which derives the overall anomaly score always from the context, in which the sample H is the most normal.

Hypersphere Collapse

As analyzed in Chapter 2 (see Section 2.2.2) and also observed in deep clustering [66], neural approaches that aim to learn a concentrated representation of the data are (in absence of any additional constraints or regularization; see Section and 2.2.3) prone to converge to trivial, constant solutions. In theory, CVDD may also suffer from such a trivial hypersphere collapse. There exists an optimal solution (C^*, W_1^*, W_2^*) for which

the (nonnegative) CVDD objective (3.7) becomes zero due to trivial representations. This is the case for (C^*, W_1^*, W_2^*) where

$$\boldsymbol{m}_k(H^{(i)}; W_1^*, W_2^*) = \boldsymbol{c}_k^* \qquad \forall i \in \{1, \ldots, n\}, \tag{3.14}$$

holds, that is, if all contextual embeddings $\boldsymbol{m}_k(\,\cdot\,; W_1^*, W_2^*)$ have collapsed to the respective context vectors $\boldsymbol{c}_k^*$ for all input sentences S with embedding H. However, since the pre-trained embeddings H are frozen, and the self-attention embedding must be a convex combination of the columns in H, this is difficult for the network to achieve in practice, given that the training corpus is sufficiently diverse. A trivial solution might only be achieved in the unlikely case that the same word occurs in all training examples. Note that such a single word model of a corpus would be successfully compact in terms of the objective, but such "Clever Hans" behavior (see Section 4.4.2) of course does not generalize well and thus is mostly undesirable in practice. Lastly, note that as the contextual embeddings $\boldsymbol{m}_k$ and context vectors $\boldsymbol{c}_k$ are normalized to be on the unit sphere in $\mathbb{R}^d$ (via the use of cosine similarity), a trivial collapse to the origin (with $m_k = \mathbf{0}$ or $c_k = \mathbf{0}$) is also not attainable.

3.2.2 Experimental Evaluation

We evaluate the performance of CVDD quantitatively in one vs. rest experiments on the Reuters-21578 [146] and 20 Newsgroups [452] datasets as well as qualitatively in an application to IMDB Movie Reviews [346] on detecting anomalous reviews. We conduct these experiments to examine the use of learning multiple, contextual representations based on pre-trained embeddings of words for anomaly detection. Moreover, we would like to investigate whether CVDD allows one to extract separate, coherent themes from an unlabeled text corpus.[2]

General Setup

Pre-trained Models We use the pre-trained GloVe [422] and fastText [67, 261] word embeddings in our experiments. For GloVe, we consider the 6B tokens vector embeddings with $d = 300$ dimensions which have been trained on the Wikipedia and Gigaword 5 corpora. For fastText, we consider the English word vectors also with $d = 300$ dimensions which have been trained on the Wikipedia and English webcrawl. We have also experimented with dynamic word embeddings from the BERT language model [135], but did not observe any improvements over GloVe or fastText on the considered datasets that would justify the additional computational cost.

Baselines We consider three baselines for aggregating word embeddings into fixed-length sentence embeddings: (i) mean, (ii) tf-idf weighted mean, and (iii) max-pooling. It has been repeatedly observed that a simple mean sentence embedding proves to be a strong baseline on many tasks [580, 28]. The tf-idf weighted mean is a natural

[2] A `PyTorch` implementation of CVDD and code to reproduce the results is available at: `https://github.com/lukasruff/CVDD-PyTorch`

extension that accounts for document-to-term co-occurrence statistics. Max-pooling is commonly applied for aggregating sequences of hidden activations [314]. For anomaly detection, we then consider a OC-SVM [496] with cosine kernel (which in this case is equivalent to SVDD [538]) used on the sentence embeddings (i)–(iii), where we always train for hyperparameters $\nu \in \{0.05, 0.1, 0.2, 0.5\}$ and report the best result.

CVDD Details We use self-attention with $d_a = 150$ for CVDD and present results for $r \in \{3, 5, 10\}$ attention heads. We use Adam [276] with a batch size of 64 for optimization and first train for 40 epochs with a learning rate of $\eta = 0.01$ after which we train for 60 epochs with $\eta = 0.001$, i.e. we establish a simple two-phase learning rate schedule. For weighting contexts, we consider the case of equal weights ($\alpha = 0$) as well as a logarithmic annealing strategy $\alpha \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ where we update α every 20 epochs. For multi-context regularization, we choose $\lambda \in \{1, 10\}$.

Data Pre-Processing On all three datasets, we always lowercase text and strip punctuation, numbers, as well as redundant whitespace. Moreover, we remove stopwords using the stopwords list from the `nltk` library [59] and only consider words with a minimum length of 3 characters.

One-Class Classification of News Articles

Setup We consider one vs. rest setups on the Reuters-21578 and 20 Newsgroups topic classification datasets to quantitatively evaluate the detection performance via the AUC measure, using the ground-truth labels in testing. That is, in every setup, one of the classes is considered the normal class and the remaining classes are considered anomalous. For the multi-label Reuters dataset, we consider the subset of samples which have one label and only carry out experiments on the classes which have at least 100 training examples remaining. For 20 Newsgroups, we consider the six top-level subject matter groups `computer`, `recreation`, `science`, `miscellaneous`, `politics`, and `religion` as distinct classes. The resulting classes and dataset sizes are reported in Table 3.3. We train the models only on the training data from the respective normal class and then test on the combined test set including all classes (with the respective non-normal classes labeled as anomalous).

Table 3.3: Dataset sizes per class on Reuters and 20 Newsgroups.

	class	#train	#test
Reuters	earn	2 840	1 083
	acq	1 596	696
	crude	253	121
	trade	250	76
	money-fx	222	87
	interest	191	81
	ship	108	36
20 Newsgroups	comp	2 857	1 909
	rec	2 301	1 524
	sci	2 311	1 520
	misc	577	382
	pol	1 531	1 025
	rel	1 419	939

Results The results are presented in Table 3.4. Overall, we can see that CVDD yields a competitive detection performance, when using the mean anomaly score

Table 3.4: Mean AUC (in %) detection performance for the one vs. rest experiments on Reuters and 20 Newsgroups.

		GloVe							fastText						
		OC-SVM			CVDD				OC-SVM			CVDD			
		mean	tf-idf	max	r=3	r=5	r=10	$\boldsymbol{c}^*$	mean	tf-idf	max	r=3	r=5	r=10	$\boldsymbol{c}^*$
Reuters	earn	91.1	88.6	77.1	**94.0**	92.8	91.8	**97.6**	87.8	82.4	74.9	**95.3**	92.7	93.9	94.5
	acq	**93.1**	77.0	81.4	90.2	88.7	91.5	**95.6**	91.8	74.1	80.2	91.0	90.3	**92.7**	92.4
	crude	92.4	90.3	91.2	89.6	92.5	**95.5**	89.4	93.3	90.2	84.7	90.9	94.1	**97.3**	85.0
	trade	99.0	96.8	93.7	98.3	98.2	**99.2**	97.9	97.6	95.0	92.1	97.9	98.1	**99.3**	97.7
	money-fx	**88.6**	81.2	73.6	82.5	76.7	82.8	**99.7**	80.5	**82.6**	73.8	**82.6**	79.8	82.5	**99.5**
	interest	97.4	93.5	84.2	92.3	91.7	**97.7**	**98.4**	91.6	88.7	82.8	93.3	92.1	95.9	**97.4**
	ship	91.2	93.1	86.5	**97.6**	96.9	95.6	**99.7**	90.0	90.6	85.0	**96.9**	94.7	96.1	**99.7**
20 Newsgroups	comp	**82.0**	81.2	54.5	70.9	66.4	63.3	**86.6**	77.5	**78.0**	65.5	74.0	68.2	64.2	**88.2**
	rec	73.2	**75.6**	56.2	50.8	52.8	53.3	68.9	66.0	**70.0**	51.9	60.6	58.5	54.1	**85.1**
	sci	60.6	**64.1**	53.0	56.7	56.8	55.7	61.0	61.0	**64.2**	57.0	58.2	57.6	55.9	**64.4**
	misc	61.8	63.1	54.1	**75.1**	70.2	68.6	**83.8**	62.3	62.1	55.7	**75.7**	70.3	68.0	**83.9**
	pol	72.5	**75.5**	64.9	62.9	65.3	65.1	75.4	73.7	**76.1**	68.1	71.5	66.4	67.1	**82.8**
	rel	78.2	**79.2**	68.4	76.3	72.9	70.7	**87.3**	77.8	**78.9**	73.9	78.1	73.2	69.5	**89.3**

over the contextual anomaly scores as defined in (3.13). We also found the CVDD performance to be robust over $\lambda \in \{1, 10\}$ and results to be similar for weighting contexts equally ($\alpha = 0$) or employing the logarithmic annealing strategy. The results in Table 3.4 present averages over these hyperparameter configurations.

Table 3.5: Top words of the CVDD model with $r = 3$ contexts from the one vs. rest experiments on the 20 Newsgroups `computer`, `politics`, and `religion` classes.

`computer`			`politics`			`religion`		
$\boldsymbol{c}_1$	$\boldsymbol{c}_2$ ($\boldsymbol{c}^*$)	$\boldsymbol{c}_3$	$\boldsymbol{c}_1$	$\boldsymbol{c}_2$	$\boldsymbol{c}_3$ ($\boldsymbol{c}^*$)	$\boldsymbol{c}_1$	$\boldsymbol{c}_2$ ($\boldsymbol{c}^*$)	$\boldsymbol{c}_3$
get	**windows**	use	kill	think	**government**	example	**god**	one
help	**software**	using	killed	know	**peace**	particular	**christ**	first
thanks	**disk**	used	escape	say	**arab**	specific	**christians**	two
appreciated	**dos**	uses	away	really	**political**	certain	**faith**	three
got	**unix**	possible	back	thing	**occupation**	analysis	**jesus**	also
know	**computer**	system	shoot	anyone	**forces**	rather	**christianity**	later
way	**hardware**	need	shot	guess	**support**	therefore	**bible**	time
try	**desktop**	allow	crying	something	**movement**	consistent	**scripture**	last
tried	**macintosh**	could	killing	understand	**leaders**	often	**religion**	year
take	**cpu**	application	fight	sure	**parties**	context	**worship**	four

To get an understanding of the theme captured by some CVDD context vector, we can look at a list of the top words for the context. We can create such lists from the words with the highest self-attention weights of the most similar sentences per context vector. Table 3.5 shows the top words lists per context for a CVDD model with $r = 3$ contexts. These lists can guide a user in selecting and weighting relevant contexts in a specific application. Following this thought, we also report the best single-context detection performance in AUC to see the benefit of contextual anomaly detection. These results are given in the $\boldsymbol{c}^*$ column in Table 3.4, which demonstrate the possible gains in performance through contextual anomaly detection. We have highlighted these best contexts in Table 3.5 and present the word lists of the best contexts for all

Table 3.6: Top words of the best single CVDD contexts $\boldsymbol{c}^*$ for the one vs. rest experiments on Reuters and 20 Newsgroups news articles.

Reuters							20 Newsgroups		
earn	acq	crude	trade	money-fx	interest	ship	rec	sci	misc
shr	acquire	oil	trade	bank	rate	port	game	use	sale
dividend	buy	crude	imports	market	pct	shipping	team	systems	offer
profit	purchase	barrels	economic	dollar	bank	ships	season	modified	shipping
qtr	acquisition	petroleum	exports	currency	rates	seamen	games	method	price
net	stake	prices	tariffs	exchange	discount	vessel	league	system	sell
prior	acquired	refinery	goods	rates	effective	canal	play	types	items
cts	assets	supply	export	liquidity	interest	cargo	win	data	sold
dividends	transaction	exports	trading	markets	lending	vessels	scoring	provide	selling
share	sell	dlr	deficit	monetary	raises	sea	playoffs	devices	brand
loss	sale	gas	pact	treasury	cuts	ferry	playoff	require	bought

other classes in Table 3.6. From these lists, we can see that the $\boldsymbol{c}^*$ contexts indeed seem to be typical for what one would expect as a characterization of the respective classes. This underlines the use of modeling multiple, contextual embeddings for anomaly detection on text. An interesting question for future research is how to systematically identify and select the most relevant contexts in a given application. Considering the word lists for interpreting the different contexts can be helpful here. Another idea would be to make use of Outlier Exposure and contrast some specific unlabeled text corpus against some larger collection of text (e.g., the Wikipedia), which should help to improve emphasizing corpus-specific themes. Finally, note that the OC-SVM applied to mean sentence embeddings also establishes a strong baseline, as has been observed on other NLP tasks.

Detecting Anomalous Movie Reviews

Setup We apply CVDD for detecting anomalous reviews in a qualitative experiment on IMDB Movie Reviews. For this, we train a CVDD model with $r = 10$ context vectors on the full IMDB train set with 25 000 movie reviews. After training, we examine the most anomalous and most normal reviews according to the CVDD anomaly scores on the IMDB test set which also includes 25 000 reviews. We use the GloVe word embeddings and otherwise keep the CVDD model configuration the same as above.

Results Table 3.7 shows the top words for each of the $r = 10$ contexts of the trained CVDD model. We can see that the different contexts indeed seem to capture different themes given in the movie reviews. Note, for example, that $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ represent positive and negative sentiments respectively, $\boldsymbol{c}_3$, $\boldsymbol{c}_7$, and $\boldsymbol{c}_{10}$ represent different aspects of cinematic language, and $\boldsymbol{c}_9$ captures names. Figure 3.8 shows the movie reviews having the highest CVDD anomaly scores and the most normal reviews w.r.t. the first three contexts $\boldsymbol{c}_1$ ("positive sentiment"), $\boldsymbol{c}_2$ ("negative sentiment"), and $\boldsymbol{c}_3$ ("plot & storyline"), i.e. the samples that have the lowest respective contextual anomaly scores. The self-attention weights here give a sample-based explanation for

Table 3.7: Top words per context on IMDB Movie Reviews for CVDD with $r = 10$ contexts.

IMDB Movie Reviews									
$\boldsymbol{c}_1$	$\boldsymbol{c}_2$	$\boldsymbol{c}_3$	$\boldsymbol{c}_4$	$\boldsymbol{c}_5$	$\boldsymbol{c}_6$	$\boldsymbol{c}_7$	$\boldsymbol{c}_8$	$\boldsymbol{c}_9$	$\boldsymbol{c}_{10}$
great	awful	plot	two	think	actions	film	head	william	movie
excellent	downright	characters	one	anybody	development	filmmakers	back	john	movies
good	stupid	story	three	know	efforts	filmmaker	onto	michael	porn
superb	inept	storyline	first	would	establishing	movie	cut	richard	sex
well	pathetic	scenes	five	say	knowledge	syberberg	bottom	davies	watch
wonderful	irritating	narrative	four	really	involvement	cinema	neck	david	teen
nice	annoying	subplots	part	want	policies	director	floor	james	best
best	inane	twists	every	never	individuals	acting	flat	walter	dvd
terrific	unfunny	tale	best	suppose	necessary	filmmaking	thick	robert	scenes
beautiful	horrible	interesting	also	actually	concerning	actors	front	gordon	flick

why a particular review is normal in a respective context. We can see that the most anomalous review is repeating the same phrase. Some of the other anomalous reviews appear to include unusual combinations of words, but otherwise we see no common anomalous patterns. Finally note that the self-attention weights only provide an explanation of words that make a sentence appear normal in a context, but do not highlight the anomalous words. Considering other ways of explaining anomaly scores on sequential networks architectures (e.g., using LRP [31]) thus would be interesting and important to study in future work.

greatest show ever mad full stop greatest show ever mad full stop greatest show ever mad full stop greatest show ever mad full stop
greatest show ever mad full stop greatest show ever mad full stop greatest show ever mad full stop greatest show ever mad full stop

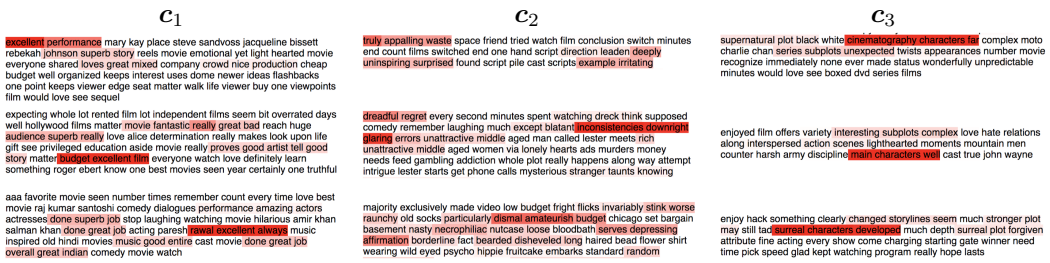
lived let tell idea heck bear walk never heard whole years really funny beginning went hill quickly

ten minutes people spewing gallons pink vomit recurring scenes enormous piles dog excrement need one say

john made two one man shows rama freaks neither one shown dvd john john put dvd john people see need see john case anyones keeping watchful aye

suspenseful subtle much much disturbing

(a) Top 5 anomalous reviews in the IMDB test set according to CVDD.



(b) Most normal reviews in the IMDB test set for CVDD contexts $\boldsymbol{c}_1$ ("positive sentiment"), $\boldsymbol{c}_2$ ("negative sentiment"), and $\boldsymbol{c}_3$ ("plot & storyline") with words highlighted by their respective self-attention weights.

Figure 3.8: Qualitative results of a CVDD model trained on IMDB Movie Reviews. The top 5 anomalous movie reviews are shown in (a). The most normal reviews w.r.t. the first three contexts with self-attention weights highlighted are shown in (b).

Conclusions from this chapter:

- FCDD introduces an explainable deep one-class classification method for anomaly detection on images, using a fully convolutional architecture to incorporate the property of spatial coherence.
- FCDD achieves state-of-the-art anomaly segmentation results in an application on detecting defects in manufacturing.
- CVDD introduces a multi-context one-class classification method for anomaly detection on text, which utilizes a multi-head self-attention mechanism to learn contextual sentence embeddings from pre-trained embeddings of words.
- CVDD can capture multiple distinct contexts given in an unlabeled text corpus and thereby enables to perform contextual anomaly detection.

Parts of this chapter are mainly based on:

[339] P. Liznerski*, L. Ruff*, R. A. Vandermeulen*, B. J. Franks, M. Kloft, and K.-R. Müller. Explainable Deep One-Class Classification. In *International Conference on Learning Representations*, 2021.

[468] L. Ruff, Y. Zemlyanskiy, R. A. Vandermeulen, T. Schnake, M. Kloft. Self-Attentive, Multi-Context One-Class Classification for Unsupervised Anomaly Detection on Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4061–4071, 2019.

4 A Unifying View of Anomaly Detection

In the previous two main chapters, Chapter 2 and Chapter 3, we have introduced deep one-class learning methods that are based on the one-class classification approach. One-class classification takes a discriminative approach to anomaly detection (see Chapter 2), seeking to learn a decision function that separates low-density from high-density regions under the normal data distribution $\mathbb{P}^+$, thereby trying to avoid a complete estimation of $\mathbb{P}^+$. This follows a theoretical formulation of anomaly detection as the problem of density level set estimation (see Section 1.3.1).

In this chapter, we present a unifying view of anomaly detection methods where we identify three main approaches: (i) one-class classification, (ii) probabilistic methods (density estimation or generative modeling), and (iii) reconstruction methods. In Section 4.1, we first discuss and analyze probabilistic methods, which aim to learn models of the normal data distribution $\mathbb{P}^+$, so that anomalies can be detected as points that have low probability under the model. In Section 4.2, we then examine reconstruction methods, which aim to learn encoding-decoding models that are optimal for normal data from $\mathbb{P}^+$, so that anomalies become apparent as points that have poor reconstruction after decoding. As we have done in Chapter 2 for one-class classification, we will show the connections between deep and shallow methods for each approach in the respective sections. We then present our unifying view in Section 4.3. We close the chapter with a comparative evaluation in Section 4.4, where we observe that the detection strategies of the various models from the different approaches are very diverse, and, using explainability techniques, find that anomaly detection models are also prone to the "Clever Hans" effect.

The unifying view presented here contributes to the long and rich history of anomaly detection research, for which there exists a wealth of review and survey literature [359, 360, 232, 570, 95, 208, 178, 516, 621, 8, 612, 427, 205, 7, 14, 443, 534, 182, 596, 571] as well as books [42, 465, 6]. There also exist some very recent surveys [90, 364, 412] that specifically focus on deep anomaly detection. However, an integrated treatment of deep learning and classic shallow methods—in particular kernel-based methods [496, 383, 538]—in the overall context of anomaly detection research has been missing prior to the publication of the review [470] that this chapter is based on.

4.1 Probabilistic Methods

As noted in the introduction (Section 1.3.1), one evident approach to density *level set* estimation is via density estimation. The category of methods we introduce in this section follows this approach, aiming to detect anomalies through estimating the normal data probability density or distribution. Many existing probability models are therefore clear candidates for the anomaly detection problem. This includes classic density estimation [212] as well as more recent deep statistical models. In the following, we describe the adaptation of these models to anomaly detection.

4.1.1 Classic Density Estimation

One of the most basic methods for multivariate anomaly detection is to compute the Mahalanobis distance from a test point to the training data mean [301]. 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 [251]. 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 kernel density estimators (KDE) [417, 212], histogram estimators, or Gaussian mixture models (GMMs) [458, 60]. KDE is arguably the most widely used nonparametric density estimator due to theoretical advantages over histograms [137] and the practical issues of GMMs with fitting and selecting hyperparameters [166]. Standard KDE, along with more recent robust variants [274, 558], is therefore a popular approach to anomaly detection. We note at this point, that a GMM with a finite number of k mixtures can also be viewed as a soft (probabilistic) clustering method that assumes k prototypical modes, which connects GMMs to prototypical reconstruction methods (see Section 4.2.1). This illustrates that there also exist connections between the three main approaches to anomaly detection. GMMs have been used, for example, to represent typical states of a machine in predictive maintenance [20].

While classic nonparametric density estimators perform fairly well for low dimensional problems, they suffer in high dimensions 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.

4.1.2 Deep Generative Models

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 $\mathbb{P}^+$. That is, their objective is to train a network ϕ_ω with weights ω such 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 the neural network ϕ_ω.

The two most established neural generative models are Variational Autoencoders (VAEs) [277, 454, 278] and Generative Adversarial Networks (GANs) [186].

VAEs

Variational Autoencoders are deep latent variable models where the input $\boldsymbol{x}$ is parameterized on latent samples $\boldsymbol{z} \sim \mathbb{Q}$ via some neural network so as to learn a distribution $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ such that $p_\theta(\boldsymbol{x}) \approx p^+(\boldsymbol{x})$. A common choice 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 that $p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_\omega(\boldsymbol{z}), \boldsymbol{\sigma}^2_\omega(\boldsymbol{z})I)$. Performing maximum likelihood estimation on θ is typically intractable. To address this, an additional network $\phi_{e,\omega'}$ (the *encoder* with weights ω') is introduced to parameterize a variational distribution $q_{\theta'}(\boldsymbol{z} \mid \boldsymbol{x})$, with parameters θ' encapsulated by the output of the encoder $\phi_{e,\omega'}$, to approximate the latent posterior $p(\boldsymbol{z} \mid \boldsymbol{x})$. The full model is then optimized in a variational Bayes manner via the evidence lower bound (ELBO):

$$\max_{\theta,\theta'} \quad -D_{\mathrm{KL}}\big(q_{\theta'}(\boldsymbol{z} \mid \boldsymbol{x}) \,\|\, p(\boldsymbol{z})\big) + \mathbb{E}_{q_{\theta'}(\boldsymbol{z} \mid \boldsymbol{x})}\big[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\big]. \tag{4.1}$$

Optimization proceeds using Stochastic Gradient Variational Bayes [277]. Given a trained VAE, $p_\theta(\boldsymbol{x})$ can be estimated via Monte Carlo sampling from the prior $p(\boldsymbol{z})$ and computing $\mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})}\left[p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\right]$. Using this estimated likelihood as an anomaly score has a nice theoretical interpretation, but experiments have shown that it tends to perform worse [595, 387] than alternatively using the reconstruction probability [21], which conditions on $\boldsymbol{x}$ to estimate $\mathbb{E}_{q_{\theta'}(\boldsymbol{z} \mid \boldsymbol{x})}\big[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\big]$. The latter can also be seen as a probabilistic reconstruction model that uses a stochastic encoding and decoding process, which connects VAEs to general reconstruction-based autoencoders (see Section 4.2.3).

GANs

Generative Adversarial Networks approximate a data distribution by posing a zero-sum-game [186]. A GAN consists of two neural networks, a *generator* $\phi_\omega : \mathcal{Z} \to \mathcal{X}$ and a *discriminator* $\psi_{\omega'} : \mathcal{X} \to (0, 1)$, which are pitted against each other. 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 and thereby learns to produce samples that are similar to the data distribution. This is achieved by using an adversarial objective:

$$\min_{\omega} \max_{\omega'} \quad \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}^+}\big[\log \psi_{\omega'}(\boldsymbol{x})\big] + \mathbb{E}_{\boldsymbol{z} \sim \mathbb{Q}}\big[\log(1 - \psi_{\omega'}(\phi_\omega(\boldsymbol{z})))\big]. \tag{4.2}$$

Training is typically performed in an alternating optimization scheme which can be a notoriously delicate procedure [479]. There exist many GAN variants, for example the Wasserstein GAN [26, 201], which is frequently used for anomaly detection

methods using GANs, or StyleGAN, which has produced impressive high-resolution photorealistic images [266].

By construction, GAN models offer no direct 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 anomaly detection [476], which is conceptually close to one-class classification (see Chapter 2). Other approaches apply optimization in latent space $\mathcal{Z}$ to find a point $\tilde{\boldsymbol{z}}$ such that $\tilde{\boldsymbol{x}} \approx \phi_\omega(\tilde{\boldsymbol{z}})$ for a test point $\tilde{\boldsymbol{x}}$. In AnoGAN [488], the authors recommend to use an intermediate layer l of the discriminator, $\psi^l_{\omega'}$, 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 $\|\psi^l_{\omega'}(\tilde{\boldsymbol{x}}) - \psi^l_{\omega'}(\phi_\omega(\tilde{\boldsymbol{z}}))\|$. In AD-GAN [130], we recommend to initialize the search for a latent point multiple times to find a collection of M latent points $\tilde{\boldsymbol{z}}_1, \ldots, \tilde{\boldsymbol{z}}_M$ while also adapting the generator parameters ω_i individually for point $\tilde{\boldsymbol{z}}_i$ to improve the reconstruction. We then propose using the mean reconstruction loss as an anomaly score:

$$\frac{1}{M}\sum_{i=1}^{M}\|\tilde{\boldsymbol{x}} - \phi_{\omega_i}(\tilde{\boldsymbol{z}}_i)\|. \tag{4.3}$$

Viewing the generator as a stochastic decoder and the optimization for an optimal latent point $\tilde{\boldsymbol{z}}$ as an (implicit) encoding of a test point $\tilde{\boldsymbol{x}}$, this way of utilizing a GAN, with the reconstruction error as an anomaly score, is similar to autoencoders (see Section 4.2.3). Later adaptations of GANs for anomaly detection 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 with incorporating the reconstruction error as an anomaly score [604, 13, 489].

Normalizing Flows

Like neural generative models, normalizing flows [140, 413, 284] also attempt to map data points from a source distribution $\boldsymbol{z} \sim \mathbb{Q}$ (termed *base distribution* for flows) so that $\boldsymbol{x} \approx \phi_\omega(\boldsymbol{z})$ is distributed according to p^+. However, a distinguishing characteristic of normalizing flows is that their latent space $\mathcal{Z} \subseteq \mathbb{R}^D$ where $\mathbb{Q}$ lives has the same dimensionality D as the input space $\mathcal{X} \subseteq \mathbb{R}^D$. A normalizing flow consists of L neural network 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 advantage of preserving the dimensionality and the invertible formulation is that the probability density of $\boldsymbol{x}$ can be calculated exactly via a change of variables

$$p_{\boldsymbol{x}}(\boldsymbol{x}) = p_{\boldsymbol{z}}(\phi_\omega^{-1}(\boldsymbol{x}))\prod_{i=1}^{L}\left|\det J\phi^{-1}_{i,\omega_i}(\boldsymbol{x}_i)\right|, \tag{4.4}$$

where $\boldsymbol{x}_L = \boldsymbol{x}$ and $\boldsymbol{x}_i = \phi^{-1}_{i+1} \circ \cdots \circ \phi^{-1}_L(\boldsymbol{x})$ otherwise. Normalizing flows are typically optimized to maximize the likelihood of the training data. Evaluating the Jacobian and its determinant for each layer can be very expensive. For this reason, the layers of normalizing flows are usually designed so that the Jacobian has some nice

structure, for example being upper (or lower) triangular, so that it is not necessary to compute the full Jacobian to evaluate its determinant [140, 141, 240]. One benefit of normalizing flows over other neural generative models (e.g., VAEs or GANs) is that the likelihood of a point can be computed directly without any approximation for flows, while also enabling reasonably efficient sampling. Since the density $p_{\boldsymbol{x}}(\boldsymbol{x})$ can be computed exactly, normalizing flow models can be directly applied to anomaly detection [386, 577], using the negative log-likelihood as an anomaly score. Maziarka et al. [365] have recently proposed another flow-based anomaly detection model that optimizes the normalizing flow to learn a data-enclosing hypersphere of minimum volume in latent space, which connects their method to deep one-class classification (see Chapter 2.2).

One limitation of normalizing flows is that, per construction, they do not perform any dimensionality reduction, which argues against their use on data where the true (effective) dimensionality is much smaller (e.g., for images that live on a lower dimensional manifold in pixel space). For image data, it has been observed that these models can often assign high likelihood to anomalous instances [387]. Recent work suggests that one reason for this phenomenon seems to be that the likelihood in current flow models is dominated by low-level features due to their specific network architectures and inductive biases [487, 281].

4.1.3 Energy-Based Models

Besides generative models, energy based models (EBMs) are some of the earliest deep statistical models [158, 236, 304]. An EBM is a model whose density is characterized by an energy function $E_\theta(\boldsymbol{x})$ with

$$p_\theta(\boldsymbol{x}) = \frac{1}{Z(\theta)} \exp\big(-E_\theta(\boldsymbol{x})\big), \tag{4.5}$$

where $Z(\theta) = \int \exp\big(-E_\theta(\boldsymbol{x})\big)\, \mathrm{d}\boldsymbol{x}$ is the so-called *partition function* which ensures that p_θ integrates to 1. EBMs are typically trained via gradient descent, approximating the log-likelihood gradient $\nabla_\theta \log p_\theta(\boldsymbol{x})$ via Markov chain Monte Carlo (MCMC) [228] or Stochastic Gradient Langevin Dynamics (SGLD) [578, 195]. Since the partition function $Z(\theta)$ is typically intractable, EBMs mostly do not allow a direct evaluation of the density p_θ. However, the energy 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 [230] and Deep Boltzmann Machines [478] are graphical models consisting of layers of latent states followed by an observed output layer that models the training data. Here, the energy function does not only depend 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 variables allows EBMs to richly model latent probabilistic dependencies in data distributions, this makes their use for anomaly detection difficult, since the latent variables must be marginalized to recover some value related to the likelihood. Later works replaced the probabilistic latent layers with deterministic ones [393] allowing for the practical evaluation and

use of $E_\theta(\boldsymbol{x})$ as an anomaly score. This sort of model has been successfully used for deep anomaly detection [606]. Recently, EBMs have also been suggested as a framework to reinterpret deep classifiers where the energy-based training has shown to improve robustness and out-of-distribution detection performance [195]. We will discuss the connection of out-of-distribution detection to anomaly detection in our outlook in Section 5.2.2, where we see an opportunity to bridge these related lines of research.

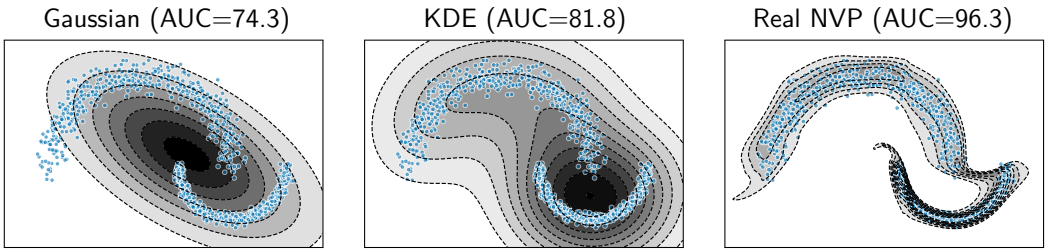


Figure 4.1: Density estimation models on the *Big Moon, Small Moon* toy example from Figure 1.4. The parametric Gaussian model is limited to an ellipsoidal (convex, unimodal) density. KDE with an RBF kernel is more flexible, yet tends to underfit the (multi-scale) distribution due a uniform kernel scale. The normalizing flow model (Real NVP [141]) is the most flexible model, yet flow architectures induce biases as well, here a connected support caused by affine coupling layers in Real NVP.

In the above discussion of probabilistic distribution and density models, we have focused on the case of i.i.d. samples of low-dimensional data and images. For comparison, we show in Figure 4.1 three canonical density estimation models (Gaussian, KDE, and Real NVP) 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 also exist many deep statistical models for other settings. For contextual anomaly detection, for example, there exist GAN [374], VAE [530], and normalizing flow [3] variants for modeling conditional distributions. Likewise there exist many deep generative models for virtually all data types including time series [530, 321], text [73, 104], or graphs [258, 68, 325], all of which potentially may be used for anomaly detection.

4.2 Reconstruction Methods

Models that are trained on a reconstruction objective are among the earliest [252, 217] and most common [90, 412] neural network-based approaches to anomaly detection. Reconstruction methods learn a model that is optimized to reconstruct normal data instances well, thereby aiming to detect anomalies by failing to accurately reconstruct them under the learned model. Most reconstruction methods have a purely geometric motivation (e.g., PCA or deterministic autoencoders), yet some probabilistic variants exist that show a connection to the problem of density (level set) estimation. In this

section, we define the general reconstruction learning objective, highlight common underlying assumptions, and present standard reconstruction methods as well as discuss recent variants.

4.2.1 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 $\mathcal{X} \subseteq \mathbb{R}^D$ 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 decoder. For reconstruction methods, the embedding $\phi_e(\boldsymbol{x}) = \boldsymbol{z}$ of a point $\boldsymbol{x}$ into latent space $\mathcal{Z} \subseteq \mathbb{R}^d$ is sometimes also called the *code* of $\boldsymbol{x}$. The reconstruction objective then is to learn ϕ_θ so 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 such that $\boldsymbol{x}$ is reconstructed with minimal error, usually measured in L^2-distance. Given unlabeled data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$, the typical reconstruction objective can be formulated as

$$\min_\theta \quad \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{x}_i - (\phi_d \circ \phi_e)_\theta(\boldsymbol{x}_i)\|^2 + \mathcal{R}, \tag{4.6}$$

where $\mathcal{R}$ again represents 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 the latent space $\mathcal{Z}$. Without any restrictions, the reconstruction objective (4.6) would be optimally solved by the identity map $\phi_\theta \equiv \mathrm{id}$, but then nothing would be learned from the data of course. In order to learn something useful, structural assumptions about the data-generating process are therefore needed. We here identify two principal assumptions: the manifold and the prototype assumptions.

Manifold Assumption

The manifold assumption asserts that the data (approximately) lives on some lower-dimensional (possibly non-linear, non-convex) manifold $\mathcal{M}$ that is embedded within the data space $\mathcal{X}$, that is, $\mathcal{M} \subset \mathcal{X}$ with $\dim(\mathcal{M}) < \dim(\mathcal{X})$. The data space $\mathcal{X}$ here is sometimes also called the *ambient* or *observation space*. For natural images observed in pixel space, for example, the manifold captures the structure of scenes as well as variation due to rotation and translation, changes in color, shape, size, texture, etc. For human voices observed in audio signal space, the manifold captures variation due to the words being spoken as well as 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 $\mathcal{Z}$ as well as functions $\phi_e : \mathcal{X} \mapsto \mathcal{Z}$ and $\phi_d : \mathcal{Z} \mapsto \mathcal{X}$ such that for all $x \in \mathcal{X}$, we have $x \approx \phi_d(\phi_e(x))$. In consequence, the data-generating distribution $\mathbb{P}$ can be represented as the push-forward through ϕ_d of some latent distribution $\mathbb{P}_Z$. Equivalently, the latent distribution $\mathbb{P}_Z$ is the push-forward of $\mathbb{P}$ through ϕ_e.

The learning objective under the manifold assumption therefore is to learn the pair of functions ϕ_e and ϕ_d such 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 unsupervised machine learning tasks such as manifold learning itself [312, 430], dimensionality reduction [285, 494, 286, 557], disentanglement [490, 340], and representation learning in general [50, 552].

Prototype Assumption

The prototype assumption asserts that there exists a finite number of prototypical elements in data space $\mathcal{X}$ that characterize the data distribution well. We can model this assumption in terms of a data-generating distribution that depends on a discrete latent categorical variable $Z \in \mathcal{Z} = \{1, \dots, k\}$ that captures some $k \in \mathbb{N}$ 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. The reconstruction objective under the prototype assumption aims to learn an encoding function that for $\boldsymbol{x} \in \mathcal{X}$ identifies an index $\phi_e(\boldsymbol{x}) = j \in \{1, \dots, k\}$ and a decoding function $j \mapsto \phi_d(j) = \boldsymbol{c}_j$ that maps to the respective jth prototype (or some prototypical distribution or mixture of prototypes more generally) such that the reconstruction error $\|\boldsymbol{x} - \boldsymbol{c}_j\|$ becomes minimal. In contrast to the manifold assumption, where we usually describe the data with some continuous mapping, under the (most basic) prototype assumption we characterize the data with a discrete set of vectors $\{\boldsymbol{c}_1, \dots, \boldsymbol{c}_k\} \subseteq \mathcal{X}$. This method of representing a data distribution with a set of prototypical vectors is also known as Vector Quantization (VQ) [329, 171].

Reconstruction Anomaly Score

A reconstruction model must extract common patterns and salient features from the data in its encoding (subject to the imposed model assumptions) in order that its decoding from the compressed latent representation achieves low reconstruction error (e.g., feature correlations and dependencies, frequent patterns, cluster structure, statistical redundancy, etc.). Assuming that the training data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X}$ includes mostly normal instances, we therefore expect a reconstruction model to produce a low reconstruction error for normal instances and a high reconstruction error for anomalies. For this reason, the anomaly score of reconstruction models is usually directly defined as the reconstruction error:

$$s(\boldsymbol{x}) = \|\boldsymbol{x} - (\phi_d \circ \phi_e)_\theta(\boldsymbol{x})\|^2. \tag{4.7}$$

For models that have learned some manifold structure or prototypical representation truthful to the normal data distribution $\mathbb{P}^+$, a high reconstruction error would thus detect off-manifold or non-prototypical instances.

Most reconstruction methods do not follow any probabilistic motivation, and a point $\boldsymbol{x}$ is deemed anomalous simply due to not conforming to its "idealized"

representation $\phi_d(\phi_e(\boldsymbol{x})) = \hat{\boldsymbol{x}}$ after the encoding and decoding process. Some reconstruction methods, however, also have probabilistic interpretations, such as PCA [543], or even are derived from probabilistic objectives such as Bayesian PCA [61] or VAEs [277]. These methods are again related to density (level set) estimation—usually making particular assumptions about some latent probabilistic structure—in the sense that a high reconstruction error indicates low density regions and vice versa.

4.2.2 Principal Component Analysis

Principal Component Analysis (PCA) is probably one of the most widely studied and used methods in statistics and machine learning. Several works have adapted PCA for anomaly detection [215, 249, 416, 511, 242, 149, 505], which can be considered the default reconstruction baseline.

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

$$\max_{W} \quad \sum_{i=1}^{n} \|W\boldsymbol{x}_i\|^2 \quad \text{s.t. } WW^\top = I. \tag{4.8}$$

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 component-wise variances [259]. 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.

From a reconstruction perspective, the objective of finding 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 $WW^\top = I$) that minimizes the mean squared reconstruction error,

$$\min_{W} \quad \sum_{i=1}^{n} \|\boldsymbol{x}_i - W^\top W\boldsymbol{x}_i\|^2 \quad \text{s.t. } WW^\top = I, \tag{4.9}$$

results in exactly the same PCA solution. So PCA optimally solves the reconstruction objective (4.6) for the case of a linear encoder $\phi_e(\boldsymbol{x}) = W\boldsymbol{x} = \boldsymbol{z}$ and transposed linear decoder $\phi_d(\boldsymbol{z}) = W^\top \boldsymbol{z}$ under the orthogonal constraint $WW^\top = I$.

For linear PCA, we can also readily identify its probabilistic interpretation [543], namely that the data distribution is generated from the linear transformation $X = W^\top Z + \varepsilon$ of a d-dimensional latent Gaussian distribution $Z \sim \mathcal{N}(\mathbf{0}, I)$, possibly with added Gaussian noise $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$, so that $\mathbb{P} \equiv \mathcal{N}(\mathbf{0}, W^\top W + \sigma^2 I)$. Maximizing the likelihood of this Gaussian over the encoding and decoding parameter W again yields PCA as the optimal solution [543]. Hence, PCA assumes the data to live on a d-dimensional ellipsoid embedded in data space $\mathcal{X} \subseteq \mathbb{R}^D$. Standard PCA therefore provides an illustrative example for a connection between density estimation and reconstruction.

Standard (linear) PCA is limited to data encodings that can only exploit linear feature correlations. Kernel PCA [494] introduced a non-linear generalization of component analysis by extending the PCA objective to non-linear kernel feature spaces, taking advantage of the "kernel trick." For a PSD kernel $k(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ with feature map $\phi_k : \mathcal{X} \to \mathcal{F}_k$, kernel PCA solves the reconstruction objective (4.9) in feature space $\mathcal{F}_k$,

$$\min_{W} \quad \sum_{i=1}^{n} \|\phi_k(\boldsymbol{x}_i) - W^\top W \phi_k(\boldsymbol{x}_i)\|^2 \quad \text{s.t. } WW^\top = I, \tag{4.10}$$

which results in an eigenvalue problem of the kernel matrix [494]. For kernel PCA, the reconstruction error can again serve as an anomaly score, which can be implicitly computed via the dual [234]. This reconstruction from linear principal components in kernel feature space $\mathcal{F}_k$ corresponds to a reconstruction from some non-linear subspace or manifold in input space $\mathcal{X}$ induced by the kernel [209]. Replacing the reconstruction $W^\top W \phi_k(\boldsymbol{x})$ in (4.10) 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 $\boldsymbol{c} = \frac{1}{n}\sum_{i=1}^{n} \phi(\boldsymbol{x}_i)$ for the L^2-distance. For RBF kernels, this prototype model is (up to a multiplicative constant) equivalent to kernel density estimation [234], which provides a link between kernel reconstruction and nonparametric density estimation methods. Finally, Robust PCA variants that account for data contamination or noise have been introduced as well [295, 395, 82, 590].

4.2.3 Autoencoders

Autoencoders are reconstruction models that use neural networks for the encoding and decoding of data. They were originally introduced during the 80s [401, 472, 41, 227] primarily as methods to perform non-linear dimensionality reduction [288, 229], yet they have also been studied for anomaly detection early on [252, 217]. Today, deep autoencoders are among the most widely adopted methods for deep anomaly detection in the literature [588, 91, 103, 433, 618, 623, 36, 106, 419, 1, 239, 185, 410, 394, 275] likely owing to their long history and easy-to-use standard variants. The standard autoencoder objective is given by

$$\min_{\omega} \quad \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - (\phi_d \circ \phi_e)_\omega(\boldsymbol{x}_i)\|^2 + \mathcal{R}, \tag{4.11}$$

which is a realization of the general reconstruction objective (4.6) 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 autoencoders is by simply mapping the data to a lower dimensional "bottleneck" code $\phi_e(\boldsymbol{x}) = \boldsymbol{z} \in \mathcal{Z}$ via the encoder network, which enforces a compression of the data and effectively limits the dimensionality of the manifold or subspace to be learned. For the case of linear networks, such an autoencoder recovers the same optimal subspace as spanned by the PCA eigenvectors [39, 402].

Apart from a "bottleneck," a number of different ways to regularize autoencoders have been introduced in the literature. Following ideas of sparse coding [404, 405, 320, 310], sparse autoencoders [352, 605] regularize the (possibly higher-dimensional, over-complete) latent code towards sparsity, for example via Lasso L^1-penalization [30]. Denoising autoencoders (DAEs) [565, 566] feed inputs corrupted with noise $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \varepsilon$ into the network which is then trained to reconstruct the original inputs $\boldsymbol{x}$. DAEs thereby provide a way to specify a noise model for ε, which has for example been applied to noise-robust acoustic novelty detection [358]. Robust deep autoencoders [618], which split the data into well-represented and corrupted parts similar to robust PCA [82], have been proposed for cases when the training data is assumed to be polluted with noise or unknown anomalies. Contractive autoencoders (CAEs) [456] 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 autoencoder, for example by imposing some degree of smoothness or introducing invariances towards certain types of input corruptions or transformations [239]. Therefore, these regularization choices should reflect the specific assumptions about a given domain and application.

Besides the deterministic autoencoders above, probabilistic autoencoders have also been proposed, which establish a connection to density estimation. The most explored class of probabilistic autoencoders are VAEs [277, 454, 278], as introduced in Section 4.1.2 through the lens of neural generative models, which approximately maximize the data likelihood (or evidence) by maximizing the ELBO. From a reconstruction perspective, VAEs adopt a stochastic autoencoding process that is realized by an encoding and decoding of distribution parameters with the encoder and decoder networks (e.g., mean and variance of a Gaussian), from which the latent code and reconstruction can then be sampled. For a standard Gaussian VAE, for example, with $q(\boldsymbol{z} \,|\, \boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_{\omega'}(\boldsymbol{x}), \operatorname{diag}(\boldsymbol{\sigma}_{\omega'}(\boldsymbol{x})^2))$, $p(\boldsymbol{z}) \sim \mathcal{N}(\mathbf{0}, I)$, and $p(\boldsymbol{x} \,|\, \boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{\mu}_{\omega}(\boldsymbol{z}), I)$ with encoder $\phi_{e,\omega'}(\boldsymbol{x}) = (\boldsymbol{\mu}_{\omega'}(\boldsymbol{x}), \boldsymbol{\sigma}_{\omega'}(\boldsymbol{x}))$ and decoder $\phi_{d,\omega}(\boldsymbol{z}) = \boldsymbol{\mu}_{\omega}(\boldsymbol{z})$, the empirical ELBO objective (4.1) becomes

$$\begin{aligned} \min_{\omega,\omega'} \quad \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{M} \Big[& \tfrac{1}{2} \|\boldsymbol{x}_i - \boldsymbol{\mu}_{\omega}(\boldsymbol{z}_{ij})\|^2 \\ & + D_{\mathrm{KL}}\Big(\mathcal{N}(\boldsymbol{z}_{ij}; \boldsymbol{\mu}_{\omega'}(\boldsymbol{x}_i), \operatorname{diag}(\boldsymbol{\sigma}_{\omega'}(\boldsymbol{x}_i)^2)) \,\Big\|\, \mathcal{N}(\boldsymbol{z}_{ij}; \mathbf{0}, I)\Big) \Big], \end{aligned} \tag{4.12}$$

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 $\boldsymbol{x}_i$. That is, a Gaussian VAE is trained to minimize the mean reconstruction error over samples from an encoded Gaussian that is regularized to be close to a standard isotropic Gaussian in latent space. VAEs have been used in various forms for anomaly detection [21, 595, 600], for instance on multimodal sequential data with LSTMs in robot-assisted feeding [414] and for new physics mining at the Large Hadron Collider [89]. Other probabilistic autoencoders that have been applied to anomaly detection are Adversarial Autoencoders (AAEs) [353, 433, 106]. By

adopting an adversarial loss to match and regularize the latent encoding distribution, AAEs can employ any arbitrary prior $p(\boldsymbol{z})$, so long as sampling is feasible.

Finally, other variants of autoencoders that have been applied to anomaly detection include RNN-based autoencoders [355, 344, 280, 273], convolutional autoencoders [419, 609], autoencoder ensembles [103, 273] and variants that constrain the gradients [297] or actively control the latent code topology [233] of an autoencoder. Autoencoders also have been used in hybrid, two-step approaches which utilize autoencoders for dimensionality reduction and apply traditional methods to the learned embeddings [155, 17, 483].

In Figure 4.2, we show a comparison of the manifolds learned by three canonical reconstruction models from above (PCA, kPCA, and AE) trained on the *Big Moon, Small Moon* toy data set. Each model uses a different feature representation (raw input, kernel, and neural network), resulting in different manifolds being learned.

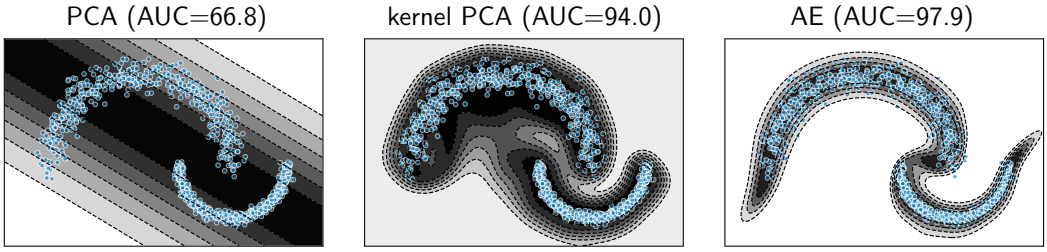


Figure 4.2: Reconstruction models on the *Big Moon, Small Moon* toy example from Figure 1.4. PCA finds the linear subspace with the lowest reconstruction error under an orthogonal projection of the data. Kernel PCA (kPCA) solves (linear) component analysis in kernel feature space which enables an optimal reconstruction from (kernel-induced) non-linear components in input space. An autoencoder (AE) with one-dimensional latent code learns a one-dimensional, non-linear manifold in input space having minimal reconstruction error.

4.2.4 Clustering Models

Clustering methods that make the prototype assumption provide another approach to reconstruction-based anomaly detection. 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 $\mathbb{P}^+$. Prototypical clustering methods [250] include well-known Vector Quantization (VQ) algorithms such as k-means, k-medians, and k-medoids, which define a Voronoi partitioning [568, 569] over the metric space where they are applied—typically the input space $\mathcal{X}$. Kernel extensions of k-means have also been studied [138] and considered for anomaly detection [192]. GMMs with a finite number of k mixtures (see Section 4.1.1) have also been used for (soft) prototypical clustering. Here, the distance to each cluster is given by the Mahalanobis distance that is defined by the covariance matrix of the respective Gaussian mixture component [20].

More recently, deep learning approaches to clustering have also been proposed [591, 556, 448, 264], some also based on k-means [598], and applied to anomaly

detection [36, 87, 17]. As in deep one-class classification (see Section 2.2), a persistent question in deep clustering is how to effectively regularize against a feature map collapse [66], since cluster representations are usually also optimized to be "compact" in some way. Note that whereas for deep clustering methods the reconstruction error is commonly measured in latent space $\mathcal{Z}$, for deep autoencoders it is measured in the input space $\mathcal{X}$ after decoding. A feature collapse in latent space (i.e., a constant encoder $\phi_e \equiv \boldsymbol{c} \in \mathcal{Z}$) would thus result in a constant decoding for a (deterministic) autoencoder (the data mean at optimum), which generally is a suboptimal solution of the autoencoder onjective (4.11). For this reason, autoencoders seem less susceptible to a feature collapse.

4.3 Unifying View

In this section, we present a unifying view of the anomaly detection problem. We identify relevant modeling components that allow us to characterize many existing methods in a systematic way. Importantly, this unifying view reveals connections that show opportunities for transferring methodological or algorithmic ideas between anomaly detection methods, for example, transferring ideas from kernel-based anomaly detection to deep methods and vice versa. Figure 4.3 gives an overview of the categorization of anomaly detection methods within our unifying view.

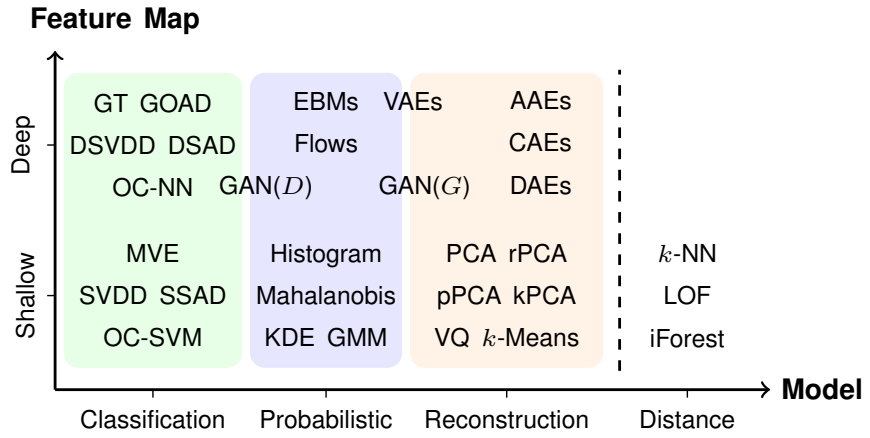


Figure 4.3: Anomaly detection approaches arranged in the plane spanned by two major components (*Model* and *Feature Map*) of our unifying view. Based on shared principles, we distinguish One-Class *Classification*, *Probabilistic* models, and *Reconstruction* models as the three main groups of approaches which all formulate *Shallow* and *Deep* models. 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 anomaly detection problem.

4.3.1 Modeling Dimensions of Anomaly Detection Methods

We identify the following five components or *modeling dimensions* for anomaly detection:

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 $f_\theta(\boldsymbol{x})$. Semi-supervised or supervised methods (see Section 1.3.2) use loss functions that incorporate labels, but for the many unsupervised anomaly detection methods we have that $\ell(s, y) = \ell(s)$. D2 **Model** defines the model f_θ that a specific method uses to map an input $\boldsymbol{x} \in \mathcal{X}$ to some scalar value that is evaluated by the loss. We have dedicated major sections of this thesis to this modeling dimension where we covered certain groups of methods that formulate models based on common principles, namely one-class classification (Chapter 2), probabilistic methods (Section 4.1), and reconstruction methods (Section 4.2). Due to the close link between anomaly detection and density estimation, as defined in the introduction and discussed throughout the thesis, many of the methods formulate a likelihood model $f_\theta(\boldsymbol{x}) = p_\theta(\boldsymbol{x} \,|\, \mathcal{D}_n)$ with negative log-loss $\ell(s) = -\log(s)$, that is, they pose a negative log-likelihood objective, where $\mathcal{D}_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ denotes the training data. Dimension D3 is the **Feature Map** $\boldsymbol{x} \mapsto \phi(\boldsymbol{x})$ that is used in a particular model f_θ. This can be an (implicit) feature map $\phi_k(\boldsymbol{x})$ induced by some kernel k for 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. Dimension D4 **Regularization** captures 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 the parameter θ here may include both, model parameters as well as feature map parameters, that is, $\theta = (\theta_f, \theta_\phi)$ in general. θ_f could be the distribution parameters of a parametric density model, for example, and θ_ϕ the weights of a neural network. The last modeling dimension D5 describes the **Inference Mode**, specifically whether a method performs Bayesian inference [541] or not.

Having identified the above modeling dimensions, we can formulate the following general anomaly detection learning objective that encompasses a broad range of anomaly detection methods:

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

For θ^* denoting a minimizer of $(*)$, we can compute the anomaly score of a test input $\tilde{\boldsymbol{x}}$ via the model $f_{\theta^*}(\tilde{\boldsymbol{x}})$. In the Bayesian case, where the objective in $(*)$ is the

negative log-likelihood of some 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 4.1, we describe many well-known anomaly detection methods identified within our unifying view.

4.3.2 Comparative Discussion

The unifying view above allows us to systematically compare anomaly detection methods. Table 4.1 shows that the probabilistic methods are largely based on a 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 1.3.1). Reconstruction methods also provide such a ranking, with the anomaly score given by the difference of a data point to its reconstruction under the model. Besides ranking and detecting anomalies, these scores make it possible to rank inliers as well, which can be used, for example, to judge cluster membership or determine prototypes (see Section 4.2.4). Reconstruction methods are particularly well suited when the data originates from some manifold or prototypical structure (see Section 4.2.1). Standard one-class classification methods, in comparison, usually do not rank inliers and aim to estimate a discriminative level set boundary (see Chapter 2). This is often incorporated into the learning objective via a hinge loss as can be seen in Table 4.1. One-class classification is generally more sample-efficient and more robust to non-representative sampling of the normal data (e.g., a sampling bias towards specific normal modes) [536], but in consequence is 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 truthfully represent in-distribution modes etc. Overall, our unifying view provides a systematic perspective on the rich diversity of anomaly detection methods and allows us to identify methodological differences. In Section 4.4.1 below, we will present a comparative empirical evaluation that includes methods from all three groups (one-class classification, probabilistic, and reconstruction) and three types of feature maps (raw input, kernel, and neural network), where we find that the detection performance in different data scenarios can be very diverse. This underlines the fact that there is no simple "silver bullet" solution to the anomaly detection problem.

Besides providing a framework for discussing and comparing methods, the unifying view also allows to identify concepts that may be transferred between shallow and deep anomaly detection methods in a systematic manner. We here discuss a few explicit examples to illustrate this point. Table 4.1 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 semi-supervised [191, 467, 469] (see also Section 2.3) or multi-sphere [192, 172, 52] model extensions provide successful examples for such a transfer. Next, observe in Table 4.1 that deep autoencoders usually consider the reconstruction error in the original data space $\mathcal{X}$ after a neural network encoding

Table 4.1: Anomaly detection methods identified with our unifying view (last column provides 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
One-Class Classification	Min. Vol. Sphere	$\max(0,s)$	$\|\boldsymbol{x}-\boldsymbol{c}\|^2-R^2$	$\boldsymbol{x}$	(input)	$(\boldsymbol{c},R)$	νR^2	✗	[536]
	Min. Vol. Ellipsoid	$\max(0,s)$	$(\boldsymbol{x}-\boldsymbol{c})^\top\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{c})-R^2$	$\boldsymbol{x}$	(input)	$(\boldsymbol{c},R,\Sigma)$	$\nu(\frac{1}{2}\|\Sigma\|_{\text{F}}^2+R^2)$	✗	[464]
	SVDD	$\max(0,s)$	$\|\phi_k(\boldsymbol{x})-\boldsymbol{c}\|^2-R^2$	$\phi_k(\boldsymbol{x})$	(kernel)	$(\boldsymbol{c},R)$	νR^2	✗	[538]
	Semi-Sup. SVDD	$\max(0,ys)$	$\|\phi_k(\boldsymbol{x})-\boldsymbol{c}\|^2-R^2$	$\phi_k(\boldsymbol{x})$	(kernel)	$(\boldsymbol{c},R)$	νR^2	✗	[538, 191]
	Soft Deep SVDD	$\max(0,s)$	$\|\phi_\omega(\boldsymbol{x})-\boldsymbol{c}\|^2-R^2$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{c},R,\omega)$	νR^2; weight decay; collapse reg. (various)	✗	[466]
	OC Deep SVDD	s	$\|\phi_\omega(\boldsymbol{x})-\boldsymbol{c}\|^2$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{c},\omega)$	weight decay; collapse reg. (various)	✗	[466]
	Deep SAD	s^y	$\|\phi_\omega(\boldsymbol{x})-\boldsymbol{c}\|^2$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{c},\omega)$	weight decay	✗	[469]
	OC-SVM	$\max(0,s)$	$\rho-\langle\boldsymbol{w},\phi_k(\boldsymbol{x})\rangle$	$\phi_k(\boldsymbol{x})$	(kernel)	$(\boldsymbol{w},\rho)$	$\nu(\frac{1}{2}\|\boldsymbol{w}\|^2-\rho)$	✗	[496]
	OC-NN	$\max(0,s)$	$\rho-\langle\boldsymbol{w},\phi_\omega(\boldsymbol{x})\rangle$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{w},\rho,\omega)$	$\nu(\frac{1}{2}\|\boldsymbol{w}\|^2-\rho)$; weight decay	✗	[92]
	Bayesian DD	$\max(0,s)$	$\|\phi_k(\boldsymbol{x})-\boldsymbol{c}\|^2-R^2$	$\phi_k(\boldsymbol{x})$	(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	[174]
	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 $\mathcal{T}=\{T_1,\dots,T_K\}$ for self-labeling	✗	[181, 222]
	GOAD (CE)	$-\log(s)$	$\prod_k\sigma_k(-\|\phi_\omega(T_k(\boldsymbol{x}))-\boldsymbol{c}_k\|^2)$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{c}_1,\dots,\boldsymbol{c}_K,\omega)$	transformations $\mathcal{T}=\{T_1,\dots,T_K\}$ for self-labeling	✗	[52]
	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	✗	[471]
	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	[348, 65]
Probabilistic Methods	Parametric Density	$-\log(s)$	$p(\boldsymbol{x}\|\theta)$	$\boldsymbol{x}$	(input)	θ	choice of density class $\{p_\theta\,\|\,\theta\in\Theta\}$	✗	[62, 385]
	Gaussian/Mahalanobis	$-\log(s)$	$\mathcal{N}(\boldsymbol{x}\|\boldsymbol{\mu},\Sigma)$	$\boldsymbol{x}$	(input)	$(\boldsymbol{\mu},\Sigma)$	–	✗	[62, 385]
	GMM	$-\log(s)$	$\sum_k\pi_k\mathcal{N}(\boldsymbol{x}\|\boldsymbol{\mu}_k,\Sigma_k)$	$\boldsymbol{x}$	(input)	$(\pi,\boldsymbol{\mu},\Sigma)$	number of mixture components K	latent	[541]
	KDE	$-\log(s)$	$\exp(-\|\phi_k(\boldsymbol{x})-\boldsymbol{\mu}\|^2)$	$\phi_k(\boldsymbol{x})$	(kernel)	$\boldsymbol{\mu}$	kernel hyperparameters (e.g., bandwidth h)	✗	[458, 60]
	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	[304, 606]
	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	✗	[413, 386]
	GAN (D-based)	$-\log(s)$	$\sigma(\langle\boldsymbol{w},\psi_\omega(\boldsymbol{x})\rangle)$	$\psi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{w},\omega)$	adversarial training	✗	[489, 477]
Reconstruction Methods	PCA	s	$\|\boldsymbol{x}-W^\top W\boldsymbol{x}\|_2^2$	$\boldsymbol{x}$	(input)	W	$WW^\top=I$	✗	[215]
	Robust PCA	s	$\|\boldsymbol{x}-W^\top W\boldsymbol{x}\|_1$	$\boldsymbol{x}$	(input)	W	$WW^\top=I$	✗	[295]
	Probabilistic PCA	$-\log(s)$	$\mathcal{N}(\boldsymbol{x}\|\boldsymbol{0},W^\top W+\sigma^2 I)$	$\boldsymbol{x}$	(input)	(W,σ^2)	linear latent Gauss model $\boldsymbol{x}=W^\top\boldsymbol{z}+\varepsilon$	latent	[543]
	Bayesian PCA	$-\log(s)$	$\mathcal{N}(\boldsymbol{x}\|\boldsymbol{0},W^\top W+\sigma^2 I)\,p(W\|\boldsymbol{\alpha})$	$\boldsymbol{x}$	(input)	(W,σ^2)	linear latent Gauss model with prior $p(W\|\boldsymbol{\alpha})$	fully	[61]
	Kernel PCA	s	$\|\phi_k(\boldsymbol{x})-W^\top W\phi_k(\boldsymbol{x})\|^2$	$\phi_k(\boldsymbol{x})$	(kernel)	W	$WW^\top=I$	✗	[494, 234]
	Autoencoder	s	$\|\boldsymbol{x}-\phi_\omega(\boldsymbol{x})\|_2^2$	$\phi_\omega(\boldsymbol{x})$	(neural)	ω	advers. (AAE), contract. (CAE), denois. (DAE), etc.	✗	[618, 275]
	VAE	$-\log(s)$	$p_{\phi_\omega}(\boldsymbol{x}\|\boldsymbol{z})$	$\phi_\omega(\boldsymbol{x})$	(neural)	ω	latent prior $p(\boldsymbol{z})$	latent	[21, 278]
	GAN (G-based)	$-\log(s)$	$p_{\phi_\omega}(\boldsymbol{x}\|\boldsymbol{z})$	$\phi_\omega(\boldsymbol{x})$	(neural)	ω	adversarial training and latent prior $p(\boldsymbol{z})$	latent	[488, 130]
	k-means	s	$\|\boldsymbol{x}-\operatorname{argmin}_{\boldsymbol{c}_k}\|\boldsymbol{x}-\boldsymbol{c}_k\|_2\|_2^2$	$\boldsymbol{x}$	(input)	$(\boldsymbol{c}_1,\dots,\boldsymbol{c}_K)$	number of prototypes K	✗	[250, 541]
	k-medians	s	$\|\boldsymbol{x}-\operatorname{argmin}_{\boldsymbol{c}_k}\|\boldsymbol{x}-\boldsymbol{c}_k\|_1\|_1$	$\boldsymbol{x}$	(input)	$(\boldsymbol{c}_1,\dots,\boldsymbol{c}_K)$	number of prototypes K	✗	[250]
	VQ	s	$\|\boldsymbol{x}-\phi_d(\operatorname{argmin}_{\boldsymbol{c}_k}\|\phi_e(\boldsymbol{x})-\boldsymbol{c}_k\|)\|$	$\phi(\boldsymbol{x})$	(various)	$(\boldsymbol{c}_1,\dots,\boldsymbol{c}_K)$	number of prototypes K	✗	[329, 171]

and decoding. Kernel PCA, in comparison, defines the error in kernel feature space $\mathcal{F}_k$. One might ask if using the reconstruction error in some neural feature space may also be useful for autoencoders, for example to shift anomaly detection towards higher-level feature spaces (see low-level sensory anomalies vs. high-level semantic anomalies in Section 1.3.1). Recent work that incorporates the reconstruction error over the hidden layers of an autoencoder [275] indeed suggests that this concept can improve semantic detection performance. Considering the reconstruction models in Table 4.1, another questions one might ask is if including the prototype assumption (see Section 4.2.1) could also be useful in deep autoencoding and how this can be done practically. The VQ-VAE model, which introduces a discrete codebook between the neural encoder and decoder, presents a way to incorporate this concept and has shown to produce reconstructions with improved quality and coherence in some settings [556, 448]. Beyond these existing proof-of-concepts for transferring ideas, which we have motivated from our unifying view here, we give further ideas to potentially explore in future research in Section 5.2.1 of the outlook.

4.3.3 Distance-based Anomaly Detection

The unifying view above focuses on anomaly detection methods that formulate a loss-based learning objective. Beyond these methods, there also exists a rich literature on purely "distance-based" anomaly detection methods and algorithms that have been extensively studied in the data mining community in particular. Many of these algorithms follow a *lazy learning* paradigm, that is there is no a priori phase of model training. Instead, these algorithms evaluate new test points with respect to the training set only as they occur. Within our unifying view, we categorize these methods as "distance-based" without further granularity, but remark that various taxonomies for these types of methods have been proposed [95, 6]. Examples of such methods include nearest-neighbor-based methods [283, 441, 213, 614, 199] such as Local Outlier Factor (LOF) [78] and tree-based partitioning methods [263] such as Isolation Forest [334, 200]. These methods usually also aim to model the high-density regions of the data in some manner, for example by scaling distances in relation to local neighborhoods of data points [78], and thus are mostly consistent with the formal probabilistic anomaly detection problem definition presented in Section 1.3. For the most part, these algorithms have been applied and studied in the original input space $\mathcal{X}$. Few of them have been considered in the context of deep learning, but there also exist some hybrid anomaly detection approaches that apply distance-based algorithms on top of deep neural embeddings obtained from pre-trained networks (e.g., [53]).

4.4 Comparative Evaluation

We complement the theoretical comparison and discussion of anomaly detection methods within our unifying view above with an empirical evaluation in this section.

For this evaluation, we consider methods from all three major approaches (one-class classification, probabilistic, and reconstruction) and three types of feature maps (raw input, kernel, and neural network). We first present the experimental setup and quantitative results in Section 4.4.1, followed by an analysis of the model detection strategies using explanation techniques in Section 4.4.2.

4.4.1 Experimental Evaluation on MNIST-C and MVTec-AD

Setup We make an empirical comparison on the synthetic MNIST-C [380] and the real-world MVTec-AD [54] dataset. MNIST-C applies a set of 15 types of corruptions to the original MNIST dataset (e.g., blurring, adding stripes, impulse noise, etc). MVTec-AD is an image dataset of anomalous defects in industrial manufacturing, which we have also considered in Section 3.1.2, and contains 15 classes that take the form of textures (e.g., carpet, wood, etc.) or objects (e.g., toothbrush, screw, etc.). For MNIST-C, we train the models on the standard (non-corrupted) MNIST training set and then test on each corruption separately. For MVTec-AD, we train distinct models on each of the 15 (defect-free) class training sets and test on the corresponding test set. We measure anomaly detection performance using the AUC measure. For this evaluation, we also include the results measured in Average Precision (AP) in Appendix C.4, where we observe the same trends as for AUC, since the MNIST-C and MVTec-AD test sets are not highly imbalanced.

Methods We compare a selection of canonical anomaly detection methods from the three major approaches (one-class classification, probabilistic, and reconstruction) and three types of feature representation (raw input, kernel, and neural network). For one-class classification, we consider Minimum Volume Ellipsoid (MVE), kernel SVDD, and Deep SVDD. From the probabilistic methods, we consider a parametric Gaussian, Kernel Density Estimation (KDE), and AnoGAN. From the reconstruction methods, we consider PCA, kernel PCA, and a standard "bottleneck" autoencoder. We give the training details of each model in Appendix B.3.

Results The results for all anomaly detection methods on MNIST-C and MVTec-AD are shown in Tables 4.2 and 4.3 respectively.

A first striking observation is the heterogeneity in performance of the various methods on the different corruption and defect classes. For example, AnoGAN performs generally well on MNIST-C but is systematically outperformed by the Deep SVDD model on MVTec-AD. Further note that the more complex deep models are not better on every class, and simpler shallow models occasionally outperform their deeper counterparts. For instance, a 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`. Overall, we observe an advantage for the deep models on the more complex, semantic detection tasks. These results underline the diversity in modeling structure of the various anomaly detection methods.

Table 4.2: Mean AUC (in %) detection performance (over 5 seeds) on MNIST-C.

	Gaussian	MVE	PCA	KDE	SVDD	kPCA	AGAN	DSVDD	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**

Table 4.3: Mean AUC (in %) detection performance (over 5 seeds) on MVTec-AD.

		Gaussian	MVE	PCA	KDE	SVDD	kPCA	AGAN	DSVDD	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

One limitation of a purely quantitative view of detection performance is that we do not gain insight into the detection strategy and generalization properties of a particular model. For this reason, we will make use of techniques for explaining various anomaly detection methods next.

4.4.2 The "Clever Hans" Effect in Anomaly Detection

In the following, we augment anomaly scores with explanations obtained from using Layer-wise Relevance Propagation (LRP) [37], which allows us to compare KDE, Deep SVDD, and AE with the same explanation technique for a consistent comparison of detection strategies.

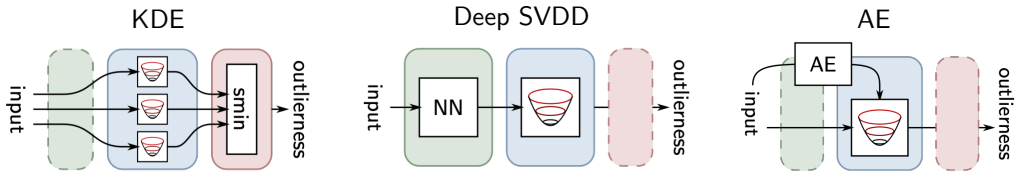


Figure 4.4: An illustration of the "neuralization" concept that reformulates existing models as strictly equivalent neural networks. Here, Kernel Density Estimation (KDE), Deep SVDD, and Autoencoder (AE) are expressed as a three-layer architecture: (i) feature extraction → (ii) distance computation → (iii) pooling. The "neuralized" formulation enables to apply LRP [37] for explaining anomalies.

Explaining Anomalies via "Neuralization" and LRP

The idea of "neuralization" is to convert non-neural network models into functionally equivalent neural networks, thereby enabling existing neural network explanation techniques (e.g., LRP) to be applied to these models [268, 267]. Figure 4.4 shows an illustration of the "neuralized" KDE, Deep SVDD, and AE models. These can be organized into a three-layer architecture, starting with (i) feature extraction, followed by (ii) distance computation, and (iii) pooling, from left to right. Once a model has been converted to a neural network structure, we can apply LRP [37] to produce an explanation of the anomaly scores. The LRP algorithm takes the score at the output of the model, propagates to "winners" in the pool, then assigns the score according to the directions in the input or feature space that contribute the most to the distance, and (if necessary) propagates the signal further down the feature hierarchy. From this, we finally obtain relevance scores for all input features (explanation heatmaps in the case of images) which highlight the features that are relevant and contribute to the anomaly score. We refer to [268] and [269] for further details.

Figure 4.5 shows the resulting explanation heatmaps of the KDE, Deep SVDD, and AE models for an example on MNIST-C `stripe`. Note that all three methods achieve a perfect AUC of 100% on this class (see Table 4.2). As we can observe from the explanations, the detection strategies of the three models are however quite different. For this reason, we expect their generalization properties (towards anomalies outside the test set) to be very different as well.

"Clever Hans" Anomaly Detectors

The MNIST-C and MVTec-AD datasets both provide ground-truth anomaly heatmaps, which makes them well-suited testbeds to assess the reliability of model detection strategies. For MNIST-C, we can create ground-truth anomaly heatmaps by computing the difference between the original and the corrupted images. The MVTec-AD dataset comes with annotated ground-truth anomaly maps of the defects. Ideally, we would like a model to base its score on the actual anomaly, that is, that the ground-truth anomaly heatmaps and explanations coincide. Here, we examine a potential discrepancy between detection performance and explanation accuracy.

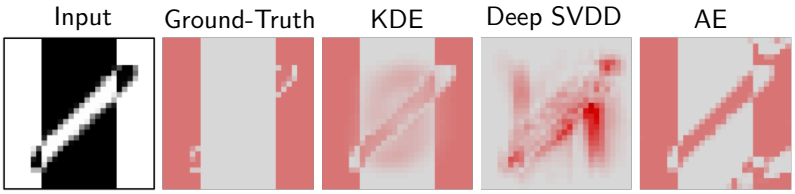


Figure 4.5: An example of LRP anomaly explanations. The input is an anomalous digit 1 from MNIST-C `stripe` that has been corrupted by inverting the pixels in the left and right vertical stripes. The ground-truth explanation highlights the anomalous pixels in red. The Kernel Density Estimation (KDE), Deep SVDD, and Autoencoder (AE) models all detect the stripe anomalies accurately, but the LRP heatmaps show that their strategies are very different: KDE highlights the anomaly, but also some regions of the digit itself. DSVDD 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.

The phenomenon, that a model makes a correct decision (or prediction), but based on the "wrong" reason (or features), is known as the "Clever Hans" effect. This effect has been previously studied in supervised learning, where classifiers have been found to make correct class predictions, but based on spurious features (e.g., image watermarks being present in the images of a class) [299], just like the horse Hans, who could correctly answer arithmetic problems by reading unintended (spurious) gestures of his owner.[1]

Table 4.4: Top 3 classes with the highest "Clever Hans" scores, defined as detection performance (measured in AUC) - explanation accuracy (measured in cosine similarity), for KDE, Deep SVDD, and AE on MNIST-C and MVTec-AD.

		KDE		Deep SVDD		AE	
MNIST-C	1.	31.9	`dotted line`	59.7	`stripe`	44.2	`shear`
	2.	31.4	`zigzag`	48.4	`dotted line`	41.4	`canny edges`
	3.	31.0	`spatter`	48.0	`impulse noise`	39.7	`motion blur`
MVTec-AD	1.	62.6	`wood`	76.0	`toothbrush`	69.1	`bottle`
	2.	61.4	`grid`	75.3	`screw`	66.4	`grid`
	3.	53.7	`zipper`	74.9	`zipper`	64.8	`wood`

As a "Clever Hans" score, we can consider the difference between the detection performance (measured in AUC) and explanation accuracy (measured in cosine similarity between the ground-truth heatmap and model explanation). The greater this score (an hence discrepancy between detection and explanation accuracy) is, the more likely a model follows a Clever Hans strategy. In Table 4.4, we show the top 3 classes for the KDE, Deep SVDD, and AE models on MNIST-C and MVTec-AD. Notably, the top classes for the three models are different, highlighting the differences in their modeling structures and detection strategies. To shed light

[1] `https://en.wikipedia.org/wiki/Clever_Hans`

on these detection strategies and possible "Clever Hans" effects, we inspect the explanations of representative anomalies in some of the top 3 classes in Figure 4.6.

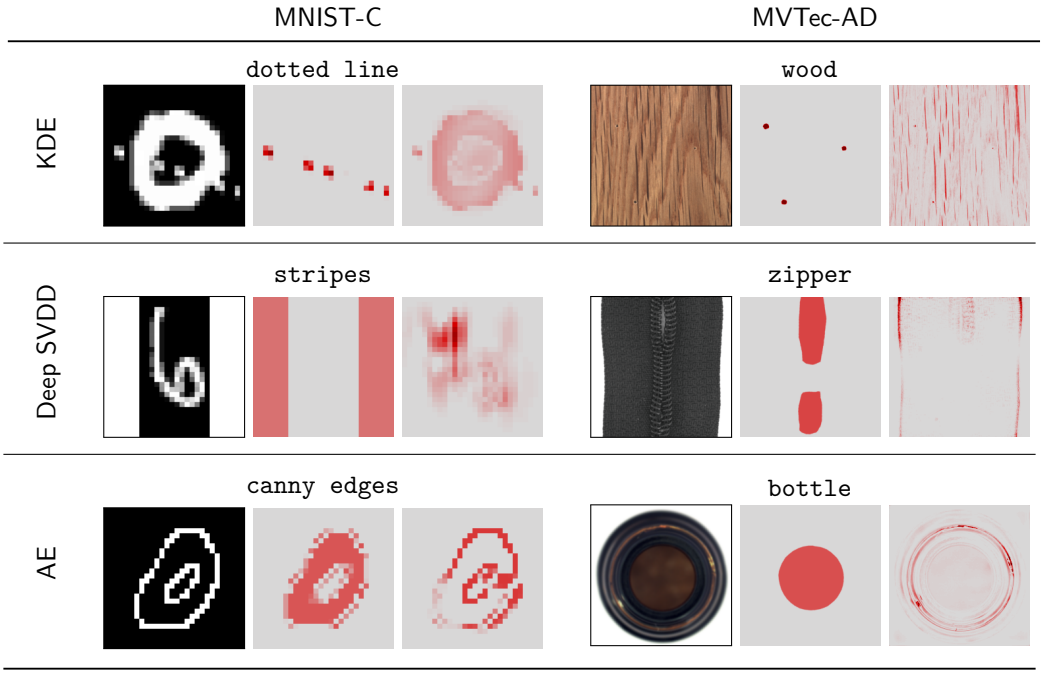


Figure 4.6: Examples taken for each model and dataset from one of the top 3 classes with highest "Clever Hans" score. For each case, we show the input image, the ground-truth explanation, and the model-based explanation, from left to right.

The KDE model on MNIST-C, for example, although correctly identifying the anomalous dotted pattern, also highlights regions of the whole digit. Similarly, for the `wood` class on MVTec-AD, the high-frequency wood grain is deemed anomalous and dominates the small local perforations in the wood panels which are the actual anomaly source. Note that KDE is the overall best performing model on the `wood` class with 94.7 AUC (see Table 4.3). However, this performance seems to be driven by a spurious correlation of a heavier wood grain appearing in the anomalous defects in the test set.

We also can observe "Clever Hans" effects for the Deep SVDD model. On MNIST-C `stripes`, the corruption turns the whole borders of the image from black to white, however, the model bases its score only on the edges of the added stripes and the interaction between these edges and the digit. On MVTec-AD `zipper`, the score is mostly based on the transition between the zipper cloth and the white background instead of the defective opening at the zipper line.

For the autoencoder model on the MNIST-C `canny edges` class, although the complete interior of the digit has turned from white to black, the autoencoder ignores this change of color and only highlights the contour of the digit. On the MVTec-AD `bottle` class, where a large contamination is given at the center of the bottle (top

view), the autoencoder bases its score on fine elements on the outer part of the bottle.

In all these examples, the anomaly detection models (correctly) yield high anomaly scores, but these scores are systematically based on the "wrong" features and not the actual ground-truth anomaly features. Clearly, such anomaly detection models that follow a "Clever Hans" strategy do not generalize well. In contrast to a supervised classification task, where there is a well-defined learning objective and the "Clever Hans" effect occurs due to the model learning to exploit spurious correlations in the data based on the training signal from labels, in the (mostly) unsupervised task of anomaly detection, the effect is critically determined by the model structure and modeling assumptions.

The "Clever Hans" effect and potential solutions to alleviate it have hardly been studied in unsupervised learning in general. In [269], we find that an ensemble of structurally different models might be one intuitive solution, by enabling multiple models to mitigate their individual structural weaknesses. Using a semi-supervised approach that includes ground-truth anomaly explanations into training, as for example possible with FCDD (see Section 3.1.2), could present another possible solution. Another practical approach is to extend the dataset (e.g., extending the `wood` data with images of normal and defective panels having a similar grain), thereby aiming to break spurious correlations. Overall, the finding that the "Clever Hans" effect occurs in anomaly detection demonstrates that an evaluation purely based on quantitative detection performance can be critically lacking or even be misleading when the available data is not representative, and highlights the importance of model interpretability and decision explanation.

Conclusions from this chapter:

- Besides one-class classification, probabilistic methods and reconstruction methods are two other main approaches to anomaly detection, which also show connections between deep and "shallow" models.
- Probabilistic methods (density estimation or generative modeling) approach anomaly detection by learning a model of the normal data distribution, thereby detecting anomalies as low probability samples.
- Reconstruction methods approach anomaly detection by optimizing an encoding-decoding model on the normal data, thereby detecting anomalies as samples with poor reconstruction after decoding.
- Our unifying view identifies five modeling components (loss, model, feature map, regularization, and inference mode) that enable to systematically characterize anomaly detection methods from all three major approaches.
- Anomaly detection models are also prone to the "Clever Hans" effect, that is, models may correctly detect anomalies, but based on the "wrong" features, and explainability techniques can be used to uncover such behavior.

Parts of this chapter are mainly based on:

[470] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller. A Unifying Review of Deep and Shallow Anomaly Detection. *Proceedings of the IEEE*, 109(5):756–795, 2021.

With added contents from:

[130] L. Deecke, R. A. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft. Image Anomaly Detection with Generative Adversarial Networks. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 3–17, 2018.

[269] J. R. Kauffmann, L. Ruff, G. Montavon, and K.-R. Müller. The Clever Hans Effect in Anomaly Detection. *preprint (under review)*, 2020.

5 Conclusion and Outlook

5.1 Conclusion

This thesis has contributed *Deep One-Class Learning*, a deep learning approach to anomaly detection. Deep one-class learning is fundamentally based on the one-class classification paradigm and extends previous methods from fixed features towards learning (or transferring) data representations via one-class learning objectives. The key idea underlying the approach is to learn a transformation (e.g., a deep neural network) that concentrates normal data in feature space, causing anomalies to be mapped away from the concentrated region, thereby making them detectable.

We have introduced the Deep SVDD method which minimizes the volume of a data-enclosing hypersphere in feature space, so that normal data points fall within and anomalies outside of the sphere. Here, we have identified a key challenge in deep one-class learning, which is the regularization against a trivial collapse solution that concentrates the feature space into a single constant. For Deep SVDD, we have analyzed this trivial solution theoretically and we have presented various ways of regularization against a collapse throughout this thesis (e.g., negative examples, adding reconstruction, inversely penalizing the embedding variance, etc.). With the Deep SAD and HSC method, we have extended Deep SVDD to the semi-supervised anomaly detection setting. In experimental evaluations we found that including few labeled ground-truth anomalies and many weakly-labeled auxiliary anomalies both can significantly improve (semantic) anomaly detection performance.

For anomaly detection on images and text, we have introduced the FCDD and the CVDD method, respectively, which incorporate specific assumptions about their respective domains into one-class learning. FCDD uses a fully convolutional architecture to incorporate the property of spatial coherence, yielding an explainable deep one-class classification method where the output directly corresponds to an anomaly heatmap. In a visual inspection application on detecting defects in manufacturing, FCDD has demonstrated state-of-the-art anomaly segmentation results. CVDD uses a multi-head self-attention mechanism to learn multiple contextual sentence embeddings based on pre-trained embeddings of words. This formulation enables contextual anomaly detection on text. In experiments we have found CVDD to extract multiple distinct themes given in an unlabeled text corpus.

Furthermore, we have presented a unifying view for a broad range of (loss-based)

anomaly detection methods, where we have identified five modeling components (loss, model, feature map, regularization, and inference mode) that allow us to systematically characterize anomaly detection methods from three major approaches (one-class classification, probabilistic methods, and reconstruction methods). For each major approach, we have established connections between their respective deep and shallow methods, as well as between the methods of the different major approaches. In a comparative evaluation, we have shown that the detection strategies of deep and shallow anomaly detection methods of the various approaches are very diverse. Using explainability techniques, we have found that anomaly detection models are also prone to the "Clever Hans" effect, that is, a model may correctly detect an anomaly, but based on the "wrong" features.

In conclusion, this thesis has demonstrated through various experiments and analyses, that the proposed deep one-class learning approach is useful for anomaly detection and can significantly improve the detection performance in various settings. In the remainder of this thesis, we turn to what lies ahead and identify specific paths for future research. In this outlook, we critically also discuss limits of the deep one-class learning approach and the results presented in this thesis, some of which we also touched on in the main chapters already.

5.2 Future Research Paths

5.2.1 Unexplored Combinations of Modeling Dimensions

The unifying view presented in Chapter 4 has exposed that a great variety of different anomaly detection methods have historically been explored along various modeling dimensions (see Figure 4.3 and Table 4.1). Our view has also shown conceptual similarities between anomaly detection methods from kernel and deep learning. The exploration of novel methods and variants has been substantially different in both learning domains, however, which offers opportunities to explore new methodology: steps that have been pursued in kernel learning but not in deep anomaly detection (or vice versa) could be transferred and powerful new solutions could emerge. In addition to existing successful examples for such a transfer (see Section 4.3.2), we here discuss a few specific ideas of so far unexplored combinations.

Consider the problem of robustness to noise and contamination. This problem has been studied well for shallow methods and there exist many effective methods [295, 77, 244, 82, 274, 590, 336]. In deep anomaly detection, very little work has addressed this specific aspect. Note that in most deep one-class learning objectives we have formulated in this thesis (see (2.2), (2.11), (2.13), and (2.20)), we have made the assumption that the training data is fairly clean by pseudo-labeling the unlabeled data as normal ($y = +1$). For HSC (see Section 2.3.5), we have found that using a pseudo-Huber loss instead of the standard L^2-norm yielded improved results on the CIFAR-10 and ImageNet one vs. rest benchmarks. We also based the FCDD method on this more robust loss formulation (see Section 3.1.1). Exploring and

analyzing such robust deep variants, possibly by taking inspiration from shallow methods (such as robust deep autoencoders [618] took inspiration from robust PCA), could prove useful for many applications. A second example is the application of Bayesian methods. Bayesian inference has been mostly considered for shallow methods [61, 174], owing to the prohibitive cost or intractability of exact Bayesian inference in deep neural networks. Recent progress in approximate Bayesian inference and Bayesian neural networks [65, 167, 298, 270, 408], however, presents the opportunity for developing methods that complement anomaly scores with uncertainty estimates or uncertainty estimates of their respective explanations [80]. For semi-supervised anomaly detection, we have successfully transferred ideas from kernel one-class classification [536, 191] to deep one-class learning in this thesis (see Sections 2.3 and 3.1.2). Probabilistic methods and reconstruction methods that can take advantage of labeled anomalies have been less explored yet. For time-series anomaly detection [165, 550, 551, 205, 302, 480], where forecasting (i.e., conditional density estimation) models are practical and widely deployed, semi-supervised 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 [224], noise contrastive estimation [207], or coding theory [559] could provide principled ways to formulate novel semi-supervised methods. Finally, active learning strategies for anomaly detection [421, 4, 526, 190], which identify informative data points for labeling, have primarily been explored for shallow detectors and could be extended to deep learning methods.

The above is certainly only a partial list of opportunities and ideas. Further analysis of methods within the unifying view will likely expose additional directions for innovation.

5.2.2 Bridging Related Lines of Research on Robustness

Other recent lines of research on robust deep learning are closely related to anomaly detection or may even be interpreted as special cases of the problem. These include research on out-of-distribution 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 ideas and transferring concepts to jointly advance research.

A basic approach to creating robust classifiers is to endow them with the ability to reject data inputs that are likely to be misclassified. This is known as *classification with a reject option* and has been studied extensively [113, 114, 43, 539, 194, 119, 170]. However, this work focuses on data points 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 data points for which no class is predicted to have high probability, following Chow's optimal rejection rule [114]. Consequently, much research has been put into developing techniques for calibrating the probabilities of classifiers [429, 202, 298, 136, 315, 396, 366] or for Bayesian uncertainty quantification

[348, 349, 65, 167, 270, 408].

Recent work has begun to address other reasons for rejecting data inputs. *Out-of-distribution (OOD) detection* considers the case where a data point is drawn from a distribution that is different from the training distribution $\mathbb{P}^+$ [219, 323, 315, 316, 110, 366]. Formally, it is impossible to determine whether an input $\boldsymbol{x}$ has been drawn from one of two distributions $\mathbb{P}_1$ and $\mathbb{P}_2$, if both distributions have support at $\boldsymbol{x}$. Hence, the OOD problem reduces to determining whether $\boldsymbol{x}$ lies outside of high density regions under $\mathbb{P}^+$, which exactly corresponds to the formal definition of anomaly detection as the problem of density level set estimation we have described in the introduction (see Section 1.3.1).

Another reason to reject a data input is because it belongs to a class that was not part of the training data. This is known as the problem of *open set recognition*. Such data can also be regarded as being generated by a distribution $\mathbb{P}^-$, so this problem also fits into the probabilistic anomaly detection framework presented in this thesis and can be addressed with all the algorithms described above. Nonetheless, researchers have developed a separate set of methods for open set recognition [485, 486, 48, 508, 335, 611]. One important future work is to compare and evaluate these methods from the anomaly detection perspective and to evaluate anomaly detection algorithms from the open set perspective.

In rejection, out-of-distribution, and open set recognition problems, there is an additional source of information that is not available in standard anomaly detection problems: the class labels of the objects. Hence, these problems combine the task of classification with anomaly detection. 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 deriving some measure to decide whether an unlabeled test point $\tilde{\boldsymbol{x}}$ should be rejected (for any of the reasons listed above). The class labels provide information about the structure of $\mathbb{P}^+$ (the in-distribution containing multiple classes), which allows to model it as a joint distribution $\mathbb{P}^+ \equiv \mathbb{P}_{X,Y}$. Methods for rejection, out-of-distribution, and open set recognition usually take advantage of this additional information. Note that the labels y here are different from the labels that mark normal or anomalous points in a supervised or semi-supervised anomaly detection setting (see Section 1.3.2).

Research on the fundamental and largely unresolved issue of adversarial examples and attacks [531, 188, 85, 549, 57, 35, 350, 86, 610, 247] is related to anomaly detection as well. We may interpret adversarial attacks as extremely hard-to-detect out-of-distribution samples [298], as they are specifically designed to target the decision boundary and confidence of a given model (usually a classifier). Standard adversarial attacks find a small perturbation δ for an input $\boldsymbol{x}$ so that $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \delta$ produces some class prediction desired by the attacker. For instance, an image of a dog may be indistinguishably perturbed from the original image to the human eye, yet the prediction of the model under attack changes from "dog" to "cat." Note that such an adversarial example $\tilde{\boldsymbol{x}}$ still likely is normal (and probably should be normal in most applications) under the data marginal $\mathbb{P}_X$ (an imperceptibly perturbed image of a dog shows a dog after all!) but the data-label pair $(\tilde{\boldsymbol{x}}, \text{"cat"})$ should be anomalous under the joint $\mathbb{P}_{X,Y}$ [101]. Methods for OOD detection have been found to also increase

the robustness against adversarial attacks and examples [298, 503, 220, 222, 110], some of which model and utilize the class conditional distributions for detection [316, 101], for the reason just described.

The above considerations highlight the relations between these lines of research towards the general goal of robust machine learning. In particular, extending deep one-class learning variants that model multiple modes [172, 52] towards incorporating the additional labeled information about the in-distribution structure given in the problems above, would be interesting to study and compare with the methods from the respective lines of research, which may help to advance these lines together. Understanding the brittleness of deep neural networks will moreover be critical for their use in anomaly detection applications that involve malicious attackers such as fraudsters or network intruders.

5.2.3 Interpretability and Trustworthiness

Much of anomaly detection research has been dedicated towards developing new methods that improve detection accuracy. In many applications, however, accuracy alone is not sufficient [88, 269] and additional criteria such as interpretability [377, 482] and trustworthiness [313, 255, 408] are equally critical, as demonstrated in Section 4.4.2. For researchers and practitioners alike [331] it is vital to understand the reasons behind the prediction or score of a particular anomaly detection model. Interpretable, explanatory feedback improves model transparency, which is indispensable for accountable decision-making [189], uncovering model failures such as "Clever Hans" effects [299, 269], and understanding model vulnerabilities that can be insightful for improving a model or system. This is especially relevant in safety-critical environments [19, 455].

Existing work on interpretable anomaly detection has considered finding subspaces of anomaly-discriminative features [369, 123, 124, 147, 567, 347], deducing sequential feature explanations [513], using feature-wise reconstruction errors [489, 54], and explaining anomalies via integrated gradients [517] or layer-wise relevance propagation [268, 269]. In the general context of the rich literature on anomaly detection though, research on interpretability and trustworthiness in anomaly detection has received relatively 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.

Anomalies might arise due to the occurrence of abnormal patterns (e.g., defects in manufacturing as in MVTec-AD [54]), but conversely also due to a lack of normal patterns. While for the former case an explanation that highlights the abnormal features is satisfactory (e.g., using heatmaps as in Sections 3.1 and 4.4.2), how should an explanation for missing normal features be conceptualized? Given the MNIST dataset of digits, for example, what should an explanation of an anomalous all-black image look like?

Recent research has also demonstrated that models and explainability techniques can be manipulated to produce explanations desired by an attacker [143, 22]. Under-

standing these vulnerabilities is also relevant for reliably explaining anomalies. An interesting question here is whether and how the vulnerability of different explainability techniques might differ. One hypothesis would be that tying an explanation directly to a model's output, which is the case for FCDD (see Section 3.1), can improve the robustness towards explanation manipulations, as an attack must target the output of the model directly (and cannot target intermediate layers), which likely influences the model prediction or anomaly score more.

The matters of interpretability and trustworthiness become more pressing as the data and applications become more complex. Effective anomaly detection solutions for complex applications necessarily require more powerful methods, for which explanations become generally harder to interpret. Therefore, future research in this direction will be imperative.

5.2.4 The Need for Challenging and Open Datasets

Challenging problems with clearly defined evaluation criteria on publicly available benchmark datasets are indispensable for measuring progress and moving a field forward. The significance of the ImageNet database [133] and corresponding challenges [474] for progressing the field of computer vision and supervised deep learning in the last decade give a prime example of this. Currently, the standard evaluation practices in deep anomaly detection [466, 13, 181, 221, 1, 424, 574, 222, 575, 469, 52, 275], out-of-distribution detection [219, 323, 315, 316, 387, 450, 110, 502], and open set recognition [485, 486, 48, 508, 335] still largely rely on repurposing classification datasets by deeming some dataset classes to be anomalous or considering in-distribution vs. out-of-distribution dataset combinations (e.g., training a model on Fashion-MNIST clothing items and regarding MNIST digits to be anomalous). We also followed this standard protocol in the experimental evaluations on the one vs. rest benchmarks presented in this thesis. Although these benchmarks have their value, it has been questioned how well they reflect real progress on challenging anomaly detection tasks [586, 471, 9]. Moreover, there is the tendency that only few methods seem to dominate most of these benchmarks in the works cited above. This is alarming, since it suggests that there may be a bias in the literature towards evaluating only the upsides of new deep learning methods, but critically leaving out an analysis of their downsides and failures (e.g., "Clever Hans" behavior). This indicates a lack of diversity in the current evaluation practices and the benchmarks being used. In the spirit of "all models are wrong" [74], more research effort should also go into studying how certain models are wrong to understand the trade-offs that they make. For instance, some models likely make a trade-off between detecting low-level vs. high-level semantic anomalies (see Section 1.3.1 and [9]). The availability of more diverse and challenging datasets would be of great benefit in this regard. Recent datasets such as MVTec-AD [54], which we have also used for evaluation in this thesis, and competitions such as the Medical Out-of-Distribution Analysis Challenge [622] are great excellent examples, but the field needs many more challenging open datasets to foster progress.

5.2.5 Weak Supervision and Self-Supervised Learning

Most anomaly detection research has been studying the problem in absence of any kind of supervision, that is, in an unsupervised setting (see Section 1.3.2). Recent work, which includes the results on utilizing few labeled true anomalies (see Section 2.3.4) or many weakly-labeled auxiliary anomalies (see Section 2.3.6) presented in this thesis, however suggests 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 the learning from imperfectly or scarcely labeled data [445, 620, 459]. Labels can be inaccurate (e.g., due to labeling errors or uncertainty) or incomplete (e.g., covering only few normal modes or specific anomalies). Current work on semi-supervised anomaly detection indicates that including even only few labeled anomalies can already yield marked performance improvements on complex data (see [411, 125, 469, 554, 471, 339] as well as Sections 2.3 and 3.1.2). The key challenge here is to formulate and optimize such methods so that they generalize well to novel anomalies. Combining these semi-supervised methods with active learning techniques can help identifying informative candidates for labeling [421, 4, 526, 190]. This can be an effective strategy for designing anomaly detection systems that continuously improve through expert feedback loops [513, 127]. Such systems have not yet been explored for deep anomaly detectors so far. Outlier exposure [221], that is, the idea of utilizing massive amounts of unlabeled data that is publicly available in some domains (e.g., stock photos for computer vision or the English Wikipedia for NLP) as auxiliary negative examples (see Section 1.3.2 and Section 2.3.1), can also be viewed as a form of weak supervision (imperfectly labeled anomalies). Though such negative examples may not coincide with ground-truth anomalies, such contrasting seems to be beneficial for learning characteristic representations of normal concepts in many domains, as also indicated by our findings in Section 2.3.6. Nedelkoski et al. [391] have applied this idea and used auxiliary log data to well characterize the normal logs of a specific computer system, but so far this has been little explored in applications. Transfer learning approaches to anomaly detection also follow the idea of distilling domain knowledge into a model, for example, through using and possibly fine-tuning pre-trained (supervised) models [409, 423, 407, 53, 269, 131]. Overall, weak forms of supervision or domain priors may prove essential for achieving effective solutions in semantic anomaly detection tasks that involve high-dimensional data, as has also been found in other unsupervised learning tasks such as disentanglement [340, 509, 341]. Hence, developing effective methods for weakly supervised anomaly detection will likely contribute to advancing the state of the art. An important questions to study in this context is whether weak forms of supervision can help to mitigate "Clever Hans" behavior (e.g., by incorporating ground-truth explanations into learning as we have done for FCDD in Section 3.1.2), or, on the other hand, also catalyze these effects.

Self-supervised learning describes the learning of data representations through solving autoregressive or auxiliary objectives, for example, next sentence or masked

words prediction [135], future frame prediction in videos [363], or the prediction of transformations applied to images [105] such as colorization [613], cropping [142, 399], or rotation [175]. Learning representations on these objectives does not require (ground-truth) labels and can thus be applied to unlabeled data, which makes self-supervised learning particularly appealing for anomaly detection. First self-supervised methods that have been introduced for visual anomaly detection train multi-class classification models based on pseudo labels that correspond to various geometric transformations (e.g., flips, translations, rotations, etc.) [181, 222, 575]. An anomaly score can then be derived from the softmax activation statistics of a so-trained classifier, assuming that anomalies exhibit higher prediction uncertainty (close to a uniform distribution). These methods have shown significant performance improvements on the common one vs. rest image benchmarks (see results reported in Section 2.3.6 and 3.1.2). Bergman and Hoshen [52] have recently proposed a generalization of this idea to non-image 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 4.1). Other very recent and promising self-supervised approaches are based on contrastive learning [105, 581, 533, 521]. One interesting aspect from the perspective of deep one-class learning here is that contrastive learning aligns and normalizes the data embeddings to lie on the unit sphere in the output space [576]. For such normalized embeddings, maximum margin hyperplane and minimum volume hypersphere separation are equivalent (which is also why the OC-SVM and SVDD are equivalent for kernels with constant norm), which may present an opportunity to unify existing methods. 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 showing that self-supervised learning helps to improve the detection of semantic anomalies and thus exhibits inductive biases towards semantic representations [9]. On the other hand, there also exists evidence showing that self-supervision mainly improves the learning of effective low-level feature representations [33]. Therefore, this research question remains to be answered, but bears great potential for many domains where large amounts of unlabeled data are available.

5.2.6 Foundation and Theory

The recent progress in anomaly detection research also raises more fundamental questions. These include open questions about the out-of-distribution generalization properties of various methods we have touched on in this thesis, the definition of anomalies in high-dimensional spaces, and information-theoretic interpretations of the problem.

Nalisnick et al. [387] have recently observed that deep generative models (DGMs) (see Section 4.1.2) such as normalizing flows or VAEs can often assign higher likelihood to anomalies than to in-distribution samples. For example, DGMs trained on Fashion-MNIST clothing items can systematically assign higher likelihood to MNIST digits

[387]. This counter-intuitive finding, which has been replicated in subsequent work [109, 221, 450, 388, 195, 502], revealed that there is a crucial lack of theoretical understanding of these models. Solidifying evidence [450, 502, 487, 281] suggests that one reason for this phenomenon seems to be that the likelihood in current DGMs is still largely biased towards 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 the context of studying this phenomenon is that for (truly) high-dimensional data, the region with highest density values must not necessarily coincide with the region of highest probability mass (also called a *typical set*), that is, the region where data points most likely occur [388]. For example, while the highest density value of a d-dimensional standard Gaussian distribution is given at the origin, data points sampled from the distribution concentrate around an annulus with radius $\sqrt{d}$ for large d [563]. Hence, data points close to the origin have the highest density values, but are unlikely to occur. This mismatch illustrates that in some scenarios (e.g., truly high-dimensional data) it might also be useful to consider other characterizations of normality in addition to a definition based on density (level sets), as given in the introduction Section 1.3. Theoretical research aimed towards understanding the above phenomenon of DGMs therefore presents an exciting research opportunity.

Similar observations indicate that reconstruction model can also systematically well reconstruct simpler out-of-distribution points, which fall within the convex hull of the training data. For example, an anomalous all-black image can be well reconstructed by an autoencoder trained on MNIST digits [547]. An even simpler example is the perfect reconstruction of data 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 the principal component in Figure 4.2). While such out-of-distribution generalization properties might be desirable for representation learning in general [292], such generalization behavior can be undesirable for anomaly detection. Therefore, more theoretical research on understanding the out-of-distribution generalization properties or biases of different methods, especially for more complex models, will be necessary.

Lastly, the development of deep anomaly detection methods also presents new opportunities to interpret and analyze the anomaly detection problem from different theoretical angles. For example, autoencoders can be understood as adhering to the Infomax principle [330, 46, 231] by implicitly maximizing the mutual information between the input and the latent code via the reconstruction objective [565]—subject to structural constraints or regularization of the code (e.g., "bottleneck," latent prior, sparsity, etc.). Similarly, as we have discussed in the information-theoretic view presented in Section 2.3.3, using Deep SVDD together with a reconstruction objective (via pre-training or as regularization), may be interpreted as following the Infomax principle with an additional "concentration" objective that the latent distribution should have minimal entropy. Park et al. [415] have presented another interpretation, connecting Deep SVDD to VAEs, showing that Deep SVDD can be seen as a special case of β-VAE that only seeks to minimize the rate (i.e., maximize compression) of

the normal data. Overall, anomaly detection has been studied comparatively less from an information-theoretic perspective [317, 237], yet information theory could prove to be fertile ground for developing a better theoretical understanding of deep one-class learning and representation learning for anomaly detection in general.

Appendix

A Ablation Studies and Sensitivity Analyses

A.1 Deep SAD Embedding Dimensionality Sensitivity Analysis

Figure A.1 shows a sensitivity analysis for Deep SAD and the hybrid SSAD method while varying the output dimensionality $d \in \{2^4, \ldots, 2^9\}$ of the network for the experimental evaluation from Section 2.3.4. We set the experimental parameters to a default of $\gamma_l = 0.05$, $\gamma_p = 0.1$, and $k_l = 1$ in this analysis, keep $\eta = 1$ for Deep SAD, and again iterate over all nine anomaly classes in every setup. We can observe that the detection performance increases and saturates with dimensionality d.

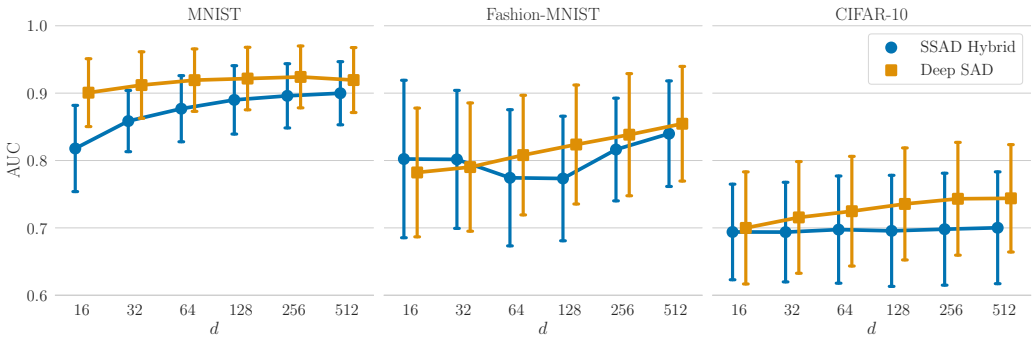


Figure A.1: Sensitivity analysis w.r.t. the network output dimensionality d for the Deep SAD method and the closest competitor hybrid SSAD. We report mean AUC with std. dev. over 90 experiments for various values of d.

A.2 Hypersphere Classifier Ablation Study

We here show an ablation for the Hypersphere Classifier (HSC) we introduced in Section 2.3.5 for varying the radial function $\rho(\boldsymbol{z}) = \exp\big(-h(\boldsymbol{z})\big)$. For this, we run the CIFAR-10 one vs. rest benchmark with 80 Million Tiny Images OE experiment

as presented in Table 2.4 in Section 2.3.6 for different functions $h : \mathbb{R}^d \to [0, \infty), \boldsymbol{z} \mapsto h(\boldsymbol{z})$. We also alter training to be with or without data augmentation in these experiments. The results are presented in Table A.1. We see that data augmentation leads to an improvement in performance even in this case where we have the full 80 Million Tiny Images dataset as OE. HSC shows the overall best performance with data augmentation and using the robust Pseudo-Huber loss $h(\boldsymbol{z}) = \sqrt{\|\boldsymbol{z}\|^2 + 1} - 1$.

Table A.1: Mean AUC (in %) detection performance (over 10 seeds) on the CIFAR-10 one vs. rest benchmark using 80 Million Tiny Images as OE for different choices of $h(\boldsymbol{z})$ in the radial function ρ of the HSC.

Data augmentation	$\|\boldsymbol{z}\|_1$	$\|\boldsymbol{z}\|_2$	$\|\boldsymbol{z}\|_2^2$	$\sqrt{\|\boldsymbol{z}\|^2 + 1} - 1$
w/o	90.6	92.3	89.1	91.8
w/	92.5	94.1	94.5	96.1

A.3 FCDD Receptive Field Sensitivity Analysis

The receptive field of FCDD (see Section 3.1.1) has an influence on both detection performance and explanation quality. Here, we provide a sensitivity analysis for varying the receptive field size of the FCDD network. We report the effects on detection performance in AUC and show qualitative heatmaps. We observe that the detection performance is only minimally affected, but larger receptive fields cause the explanation heatmaps to become less concentrated and more "blobby." On MVTec-AD, we see that this can also negatively affect the pixel-wise AUC scores, see Table A.3.

CIFAR-10 For CIFAR-10, we create eight different network architectures to study the impact of the receptive field size. Each architecture has four convolutional layers and two max-pool layers. To change the receptive field we vary the kernel size of the first convolutional layer between 3 and 17. When this kernel size is 3 then the receptive field contains approximately one quarter of the image; for a kernel size of 17 the receptive field is the entire image. Table A.2 reports the detection performance of the networks. Figure A.2 shows example heatmaps.

Table A.2: Mean AUC (over all classes and 5 seeds per class) detection performance on CIFAR-10 for neural networks with varying receptive field sizes.

Recp. field size	18	20	22	24	26	28	30	32
AUC	0.9328	0.9349	0.9344	0.9320	0.9303	0.9283	0.9257	0.9235

MVTec-AD For MVTec-AD, we create six different network architectures with different receptive field sizes. Each architecture has six convolutional layers and three

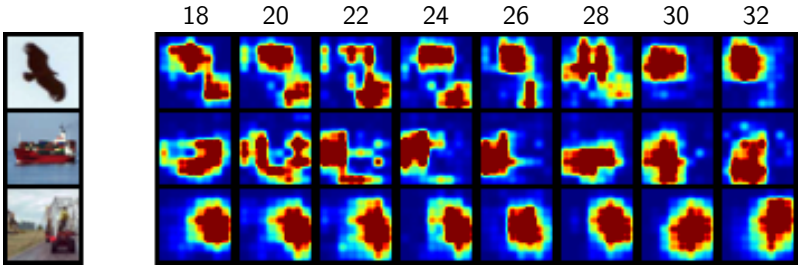


Figure A.2: Anomaly heatmaps for three anomalous test samples on CIFAR-10 models trained on normal class `airplane`. We increase the receptive field size from 18 (left) to 32 (right).

max-pool layers. We vary the kernel size for all of the convolutional layers between 3 and 13, which corresponds to a receptive field containing 1/16 of the image to the full image respectively. Table A.3 reports the explanation performance of the networks measured in pixel-wise mean AUC. Figure A.3 shows some example heatmaps. We observe that a smaller receptive field size yields better explanation performance.

Table A.3: Pixel-wise mean AUC (over all classes and 5 seeds per class) on MVTec-AD for neural networks with varying receptive field size.

Recp. field size	53	91	129	167	205	243
AUC	0.88	0.85	0.79	0.76	0.75	0.75

A.4 FCDD Gaussian Upsampling Sensitivity Analysis

FCDD can provide full-resolution anomaly heatmaps using the proposed heatmap upsampling from Section 3.1.1. This upsampling involves the choice of the variance parameter σ for the Gaussian kernel. We here analyze the effect of this hyperparameter on the explanation performance of FCDD on MVTec-AD. Table A.4 reports the pixel-wise mean AUC and Figure A.4 shows the corresponding explanation heatmaps.

Table A.4: Pixel-wise mean AUC (over all classes and 5 seeds per class) on MVTec-AD for different Gaussian upsampling variance parameters σ.

σ	4	6	8	10	12	14	16
AUC	0.8567	0.8836	0.9030	0.9124	0.9164	0.9217	0.9208

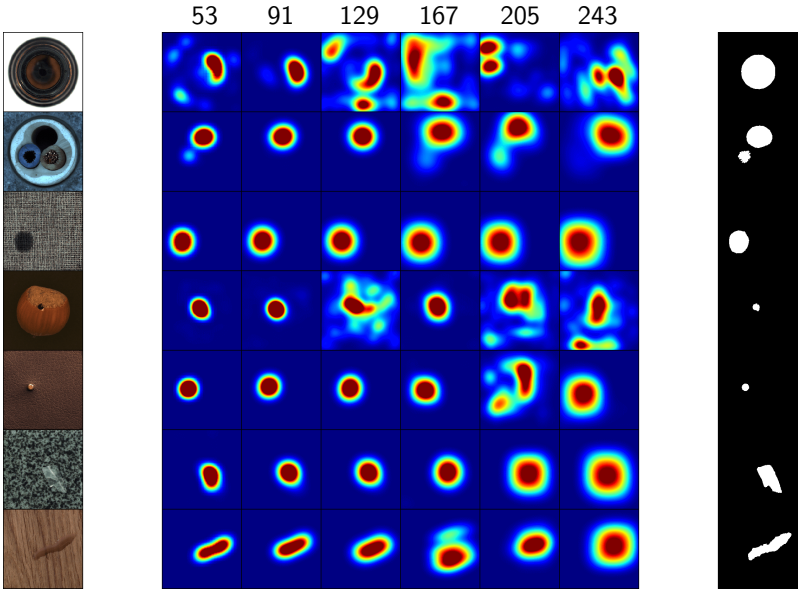


Figure A.3: Anomaly heatmaps for seven anomalous test samples of MVTec-AD. We increase the receptive field size from 53 (left) to 243 (right).

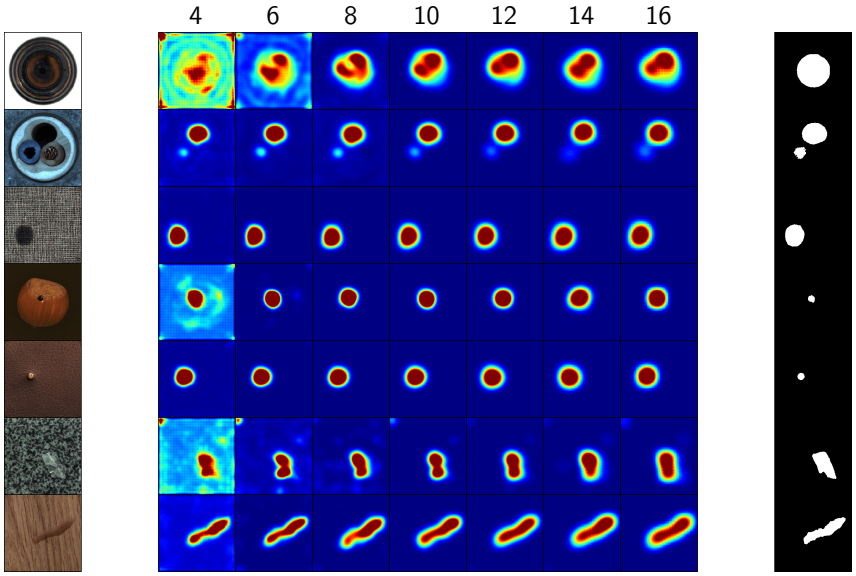


Figure A.4: Anomaly heatmaps for seven anomalous test samples of MVTec-AD. We increase the Gaussian upsampling variance parameter σ from 4 (left) to 16 (right).

B Supplementary Details

B.1 Details of Experimental Evaluation on Using Few True Anomalies

Below, we provide the network architectures and training details for all methods of the experimental evaluation on using few true anomalies from Section 2.3.4.

B.1.1 Network Architectures

We employ LeNet-type convolutional neural networks (CNNs) on MNIST, Fashion-MNIST, and CIFAR-10, where each convolutional module consists of a convolutional layer followed by leaky ReLU activations with leakiness $\alpha = 0.1$ and (2×2)-max-pooling. On MNIST, we employ a CNN with two modules, $8\times(5\times5)$-filters followed by $4\times(5\times5)$-filters, and a final fully connected layer of 32 units. On Fashion-MNIST, we employ a CNN also with two modules, $16\times(5\times5)$-filters and $32\times(5\times5)$-filters, followed by two fully connected layers of 64 and 32 units respectively. On CIFAR-10, we employ a CNN with three modules, $32\times(5\times5)$-filters, $64\times(5\times5)$-filters, and $128\times(5\times5)$-filters, followed by a final fully connected layer of 128 units.

On the classic anomaly detection benchmark datasets, we employ standard MLP feedforward architectures. On arrhythmia, a 3-layer MLP with 128-64-32 units. On cardio, satellite, satimage-2, and shuttle a 3-layer MLP with 32-16-8 units. On thyroid a 3-layer MLP with 32-16-4 units.

For the autoencoders, we always employ the above architectures for the encoder and then construct the decoder symmetrically, where we replace max-pooling and convolutions with simple upsampling and deconvolutions respectively.

B.1.2 Training Details of Competing Methods

OC-SVM/SVDD The OC-SVM and SVDD are equivalent for the Gaussian/RBF kernel we employ. As mentioned in Section 2.3.4, we deliberately grant the OC-SVM/SVDD an unfair advantage by selecting its hyperparameters to maximize AUC on a subset (10%) of the test set to establish a strong baseline. To do this, we consider the RBF scale parameter $\gamma \in \{2^{-7}, 2^{-6}, \dots 2^{2}\}$ and select the parameter

of the best performing model. Moreover, we always repeat this over ν-parameter $\nu \in \{0.01, 0.05, 0.1, 0.2, 0.5\}$ and finally report the best result.

Isolation Forest (iForest) We set the number of trees to $t = 100$ and the sub-sampling size to $\psi = 256$, as recommended in the original work [334].

Kernel Density Estimation (KDE) We select the bandwidth h of the Gaussian kernel from $h \in \{2^{0.5}, 2^1, \ldots, 2^5\}$ via 5-fold cross-validation using the log-likelihood score as in Section 2.2.4.

SSAD We also deliberately grant the semi-supervised kernel method SSAD the unfair advantage of selecting its hyperparameters optimally to maximize AUC on a subset (10%) of the test set. To do this, we again select the scale parameter γ of the RBF kernel we use from $\gamma \in \{2^{-7}, 2^{-6}, \ldots 2^2\}$ and select the parameter of the best performing model. Otherwise we set the hyperparameters as recommend by the original authors to $\kappa = 1$, $\kappa = 1$, $\eta_u = 1$, and $\eta_l = 1$ [191].

Autoencoder (AE) To create the (convolutional) autoencoders, we symmetrically construct the decoders w.r.t. the encoder architectures reported in Appendix B.1.1. Here, we replace max-pooling and convolutions with simple upsampling and deconvolutions respectively. We train the autoencoders on the MSE reconstruction loss that also serves as the anomaly score.

Hybrid Variants To establish hybrid methods, we apply the OC-SVM, IF, KDE, and SSAD as outlined above to the resulting bottleneck code embeddings given by the respective trained autoencoders.

Unsupervised Deep SVDD We consider both variants, soft-boundary Deep SVDD and One-Class Deep SVDD as unsupervised baselines and always report the better performance. For soft-boundary Deep SVDD, we optimally solve for the radius R on every mini-batch and run experiments for $\nu \in \{0.01, 0.1\}$. We set the weight decay hyperparameter to $\lambda = 10^{-6}$. For Deep SVDD, we remove the bias terms from the networks in this experiment as a measure against a trivial collapse solution.

Deep SAD We set $\lambda = 10^{-6}$ and equally weight the unlabeled and labeled examples by setting $\eta = 1$ if not reported otherwise.

SS-DGM We consider both the M2 and M1+M2 model and always report the better performing result. Otherwise we follow the settings as recommended in the original work [279]. Note that we use the latent class probability estimate (normal vs. anomalous) of the semi-supervised DGM as a natural choice for the anomaly score, and not the reconstruction error as used for unsupervised autoencoding models such as the (convolutional) autoencoder. Such deep semi-supervised models designed for

classification as the downstream task have no notion of out-of-distribution and again implicitly make the cluster assumption [493, 525, 97]. Thus, semi-supervised DGM also suffers from overfitting to previously seen anomalies similar to the supervised model which explains its poor detection performance.

Supervised Deep Binary Classifier To interpret anomaly detection as a binary classification problem, we rely on the typical assumption that most of the unlabeled training data is normal by assigning $y = +1$ to all unlabeled examples. Already labeled normal examples and labeled anomalies retain their assigned labels of $\tilde{y} = +1$ and $\tilde{y} = -1$ respectively. We train the supervised classifier on the binary cross-entropy loss. Note that in scenario (i), in particular, the supervised classifier has perfect, unpolluted label information but still fails to generalize as there are novel anomaly classes at testing time.

SGD Optimization Details for Deep Methods We use the Adam optimizer with recommended default hyperparameters [276] and apply Batch Normalization [248] in SGD optimization. For all deep approaches and on all datasets, we employ a two-stage ("searching" and "fine-tuning") learning rate schedule. In the searching phase we first train with a learning rate $\eta = 10^{-4}$ for 50 epochs. In the fine-tuning phase we train with $\eta = 10^{-5}$ for another 100 epochs. We always use a batch size of 200. For the autoencoder, SS-DGM, and the supervised classifier, we initialize the network with uniform Glorot weights [177]. For Deep SVDD and Deep SAD, we initialize the network weights ω with the weights of the encoder from a pre-trained autoencoder, i.e. we use an unsupervised autoencoder pre-training routine. We set the hypersphere center $\boldsymbol{c}$ as the mean of the network embeddings that we obtain from an initial forward pass on the data (where we exclude the labeled anomalies).

B.2 FCDD Network Architectures

We here provide the FCDD architectures used in the experiments in Section 3.1.2.

Fashion-MNIST

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 128, 28, 28]           3,328
       BatchNorm2d-2          [-1, 128, 28, 28]             256
         LeakyReLU-3          [-1, 128, 28, 28]               0
         MaxPool2d-4          [-1, 128, 14, 14]               0
            Conv2d-5          [-1, 128, 14, 14]         409,728
         MaxPool2d-6            [-1, 128, 7, 7]               0
            Conv2d-7              [-1, 1, 7, 7]             129
================================================================
Total params: 413,441
Trainable params: 413,441
Non-trainable params: 0
Receptive field (pixels): 16 x 16
----------------------------------------------------------------
```

CIFAR-10

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 128, 32, 32]           3,584
       BatchNorm2d-2          [-1, 128, 32, 32]             256
         LeakyReLU-3          [-1, 128, 32, 32]               0
         MaxPool2d-4          [-1, 128, 16, 16]               0
            Conv2d-5          [-1, 256, 16, 16]         295,168
       BatchNorm2d-6          [-1, 256, 16, 16]             512
         LeakyReLU-7          [-1, 256, 16, 16]               0
            Conv2d-8          [-1, 256, 16, 16]         590,080
       BatchNorm2d-9          [-1, 256, 16, 16]             512
        LeakyReLU-10          [-1, 256, 16, 16]               0
        MaxPool2d-11            [-1, 256, 8, 8]               0
           Conv2d-12            [-1, 128, 8, 8]         295,040
           Conv2d-13              [-1, 1, 8, 8]             129
================================================================
Total params: 1,185,281
Trainable params: 1,185,281
Non-trainable params: 0
Receptive field (pixels): 22 x 22
----------------------------------------------------------------
```

ImageNet, MVTec-AD, and Pascal VOC

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 224, 224]           1,792
       BatchNorm2d-2         [-1, 64, 224, 224]             128
              ReLU-3         [-1, 64, 224, 224]               0
         MaxPool2d-4         [-1, 64, 112, 112]               0
            Conv2d-5        [-1, 128, 112, 112]          73,856
       BatchNorm2d-6        [-1, 128, 112, 112]             256
              ReLU-7        [-1, 128, 112, 112]               0
         MaxPool2d-8          [-1, 128, 56, 56]               0
            Conv2d-9          [-1, 256, 56, 56]         295,168
      BatchNorm2d-10          [-1, 256, 56, 56]             512
             ReLU-11          [-1, 256, 56, 56]               0
           Conv2d-12          [-1, 256, 56, 56]         590,080
      BatchNorm2d-13          [-1, 256, 56, 56]             512
             ReLU-14          [-1, 256, 56, 56]               0
        MaxPool2d-15          [-1, 256, 28, 28]               0
           Conv2d-16          [-1, 512, 28, 28]       1,180,160
      BatchNorm2d-17          [-1, 512, 28, 28]           1,024
             ReLU-18          [-1, 512, 28, 28]               0
           Conv2d-19          [-1, 512, 28, 28]       2,359,808
      BatchNorm2d-20          [-1, 512, 28, 28]           1,024
             ReLU-21          [-1, 512, 28, 28]               0
           Conv2d-22            [-1, 1, 28, 28]             513
================================================================
Total params: 4,504,833
Trainable params: 4,504,833
Non-trainable params: 0
Receptive field (pixels): 62 x 62
----------------------------------------------------------------
```

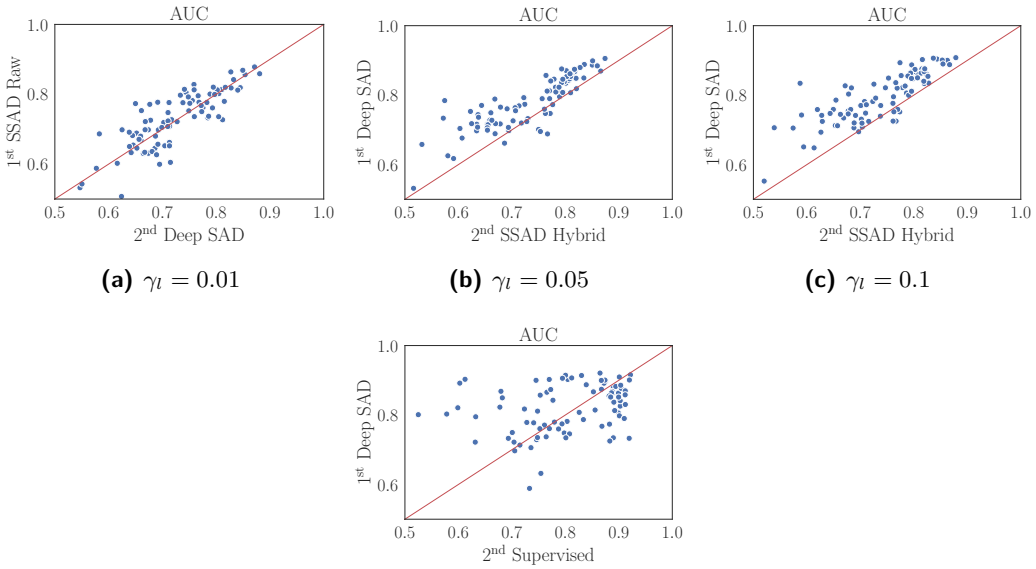
B.3 Training Details of Experiments on MNIST-C and MVTec-AD

Here we provide the training details for the methods of the experimental evaluation on MNIST-C and MVTec-AD given in Section 4.4.1. For PCA, we compute the reconstruction error whilst maintaining 90% of 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 minimum covariance determinant estimator [464] 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 the 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 SVDD one-class classifier uses a pre-trained VGG16 model (on MVTec-AD) and a CNN classifier pre-trained on the EMNIST letter subset (on MNIST-C). We apply a whitening (sphering) transform on the representations after the first fully connected layer of the pre-trained networks. 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 [239] 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 Supplementary Results

C.1 Best vs. Second Best on CIFAR-10 when Using Few True Anomalies

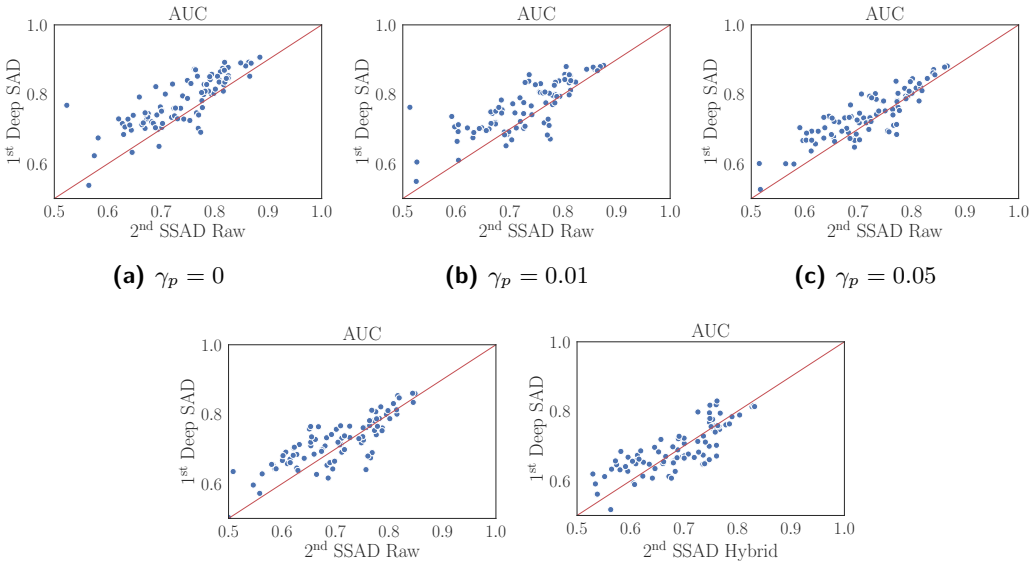
We provide AUC scatterplots in Figures C.1–C.3 of the best (1^{st}) vs. second best (2^{nd}) performing methods in the experimental scenarios (i)–(iii) from Section 2.3.4 on the most complex CIFAR-10 dataset. If most points fall above the identity line, it is a strong indication that the best method indeed significantly outperforms the second best, which often is the case for our Deep SAD method.



(a) $\gamma_l = 0.01$ **(b)** $\gamma_l = 0.05$ **(c)** $\gamma_l = 0.1$

(d) $\gamma_l = 0.2$

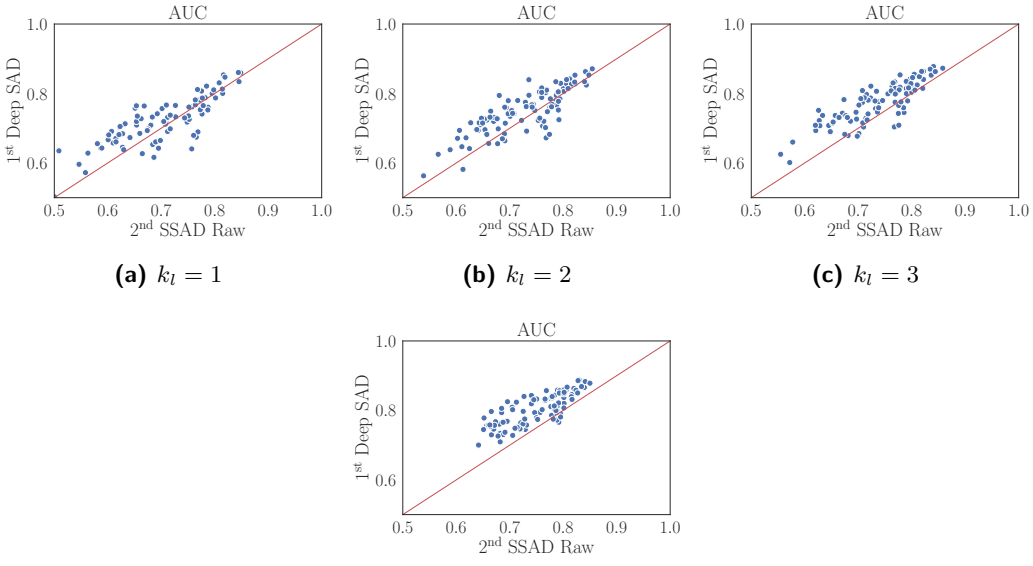
Figure C.1: AUC scatterplots of best (1^{st}) vs. second best (2^{nd}) performing methods in experimental scenario (i) on CIFAR-10, where we increase the ratio of labeled anomalies γ_l in the training set.



(a) $\gamma_p = 0$ **(b)** $\gamma_p = 0.01$ **(c)** $\gamma_p = 0.05$

(d) $\gamma_p = 0.1$ **(e)** $\gamma_p = 0.2$

Figure C.2: AUC scatterplots of best (1st) vs. second best (2nd) performing methods in experimental scenario (ii) on CIFAR-10, where we pollute the unlabeled part of the training set with (unknown) anomalies at various ratios γ_p.



(a) $k_l = 1$ **(b)** $k_l = 2$ **(c)** $k_l = 3$

(d) $k_l = 5$

Figure C.3: AUC scatterplots of best (1st) vs. second best (2nd) performing methods in experimental scenario (iii) on CIFAR-10, where we increase the number of anomaly classes k_l included in the labeled training data.

C.2 Full Results of Experimental Evaluation on Using Few True Anomalies

In Tables C.2–C.1, we provide the complete experimental results for all methods and scenarios from the experimental evaluation from Section 2.3.4.

Table C.1: Complete results on classic anomaly detection benchmark datasets in the setting with no pollution $\gamma_p = 0$ and a ratio of labeled anomalies of $\gamma_l = 0.01$ in the training set. We report the mean AUC with std. dev. computed over 10 seeds.

	OC-SVM Raw	OC-SVM Hybrid	AE	Deep SVDD	SSAD Raw	SSAD Hybrid	SS-DGM	Deep SAD	Supervised Classifier
arrhythmia	84.5±3.9	76.7±6.2	74.0±7.5	74.6±9.0	86.7±4.0	78.3±5.1	50.3±9.8	75.9±8.7	39.2±9.5
cardio	98.5±0.3	82.8±9.3	94.3±2.0	84.8±3.6	98.8±0.3	86.3±5.8	66.2±14.3	95.0±1.6	83.2±9.6
satellite	95.1±0.2	68.6±4.8	80.0±1.7	79.8±4.1	96.2±0.3	86.9±2.8	57.4±6.4	91.5±1.1	87.2±2.1
satimage-2	99.4±0.8	96.7±2.1	99.9±0.0	98.3±1.4	99.9±0.1	96.8±2.1	99.2±0.6	99.9±0.1	99.9±0.1
shuttle	99.4±0.9	94.1±9.5	98.2±1.2	86.3±7.5	99.6±0.5	97.7±1.0	97.9±0.3	98.4±0.9	95.1±8.0
thyroid	98.3±0.9	91.2±4.0	75.2±10.2	72.0±9.7	97.9±1.9	95.3±3.1	72.7±12.0	98.6±0.9	97.8±2.6

Table C.2: Complete results of experimental scenario (i), where we increase the ratio of labeled anomalies γ_l in the training set. We report the mean AUC with std. dev. computed over 90 experiments at various ratios γ_l.

	γ_l	OC-SVM Raw	OC-SVM Hybrid	iForest Raw	iForest Hybrid	KDE Raw	KDE Hybrid	AE	Deep SVDD	SSAD Raw	SSAD Hybrid	SS-DGM	Deep SAD	Supervised Classifier
MNIST	.00	96.0±2.9	96.3±2.5	85.4±8.7	90.5±5.3	95.0±3.3	87.8±5.6	92.9±5.7	92.8±4.9	96.0±2.9	96.3±2.5		92.8±4.9	
	.01									96.6±2.4	96.8±2.3	89.9±9.2	96.4±2.7	92.8±5.5
	.05									93.3±3.6	97.4±2.0	92.2±5.6	96.7±2.4	94.5±4.6
	.10									90.7±4.4	97.6±1.7	91.6±5.5	96.9±2.3	95.0±4.7
	.20									87.2±5.6	97.8±1.5	91.2±5.6	96.9±2.4	95.6±4.4
F-MNIST	.00	92.8±4.7	91.2±4.7	91.6±5.5	82.5±8.1	92.0±4.9	69.7±14.4	90.2±5.8	89.2±6.2	92.8±4.7	91.2±4.7		89.2±6.2	
	.01									92.1±5.0	89.4±6.0	65.1±16.3	90.0±6.4	74.4±13.6
	.05									88.3±6.2	90.5±5.9	71.4±12.7	90.5±6.5	76.8±13.2
	.10									85.5±7.1	91.0±5.6	72.9±12.2	91.3±6.0	79.0±12.3
	.20									82.0±8.0	89.7±6.6	74.7±13.5	91.0±5.5	81.4±12.0
CIFAR-10	.00	62.0±10.6	63.8±9.0	60.0±10.0	59.9±6.7	59.9±11.7	56.1±10.2	56.2±13.2	60.9±9.4	62.0±10.6	63.8±9.0		60.9±9.4	
	.01									73.0±8.0	70.5±8.3	49.7±1.7	72.6±7.4	55.6±5.0
	.05									71.5±8.1	73.3±8.4	50.8±4.7	77.9±7.2	63.5±8.0
	.10									70.1±8.1	74.0±8.1	52.0±5.5	79.8±7.1	67.7±9.6
	.20									67.4±8.8	74.5±8.0	53.2±6.7	81.9±7.0	80.5±5.9

Table C.3: Complete results of experimental scenario (ii), where we pollute the unlabeled part of the training set with (unknown) anomalies. We report the mean AUC with std. dev. computed over 90 experiments at various ratios γ_p.

	γ_p	OC-SVM Raw	OC-SVM Hybrid	iForest Raw	iForest Hybrid	KDE Raw	KDE Hybrid	AE	Deep SVDD	SSAD Raw	SSAD Hybrid	SS-DGM	Deep SAD	Supervised Classifier
MNIST	.00	96.0±2.9	96.3±2.5	85.4±8.7	90.5±5.3	95.0±3.3	87.8±5.6	92.9±5.7	92.8±4.9	97.9±1.8	97.4±2.0	92.2±5.6	96.7±2.4	94.5±4.6
	.01	94.3±3.9	95.6±2.5	85.2±8.8	90.6±5.0	91.2±4.9	87.9±5.3	91.3±6.1	92.1±5.1	96.6±2.4	95.2±2.3	92.0±6.0	95.5±3.3	91.5±5.9
	.05	91.4±5.2	93.8±3.9	83.9±9.2	89.7±6.0	85.5±7.1	87.3±7.0	87.2±7.1	89.4±5.8	93.4±3.4	89.5±3.9	91.0±6.9	93.5±4.1	86.7±7.4
	.10	88.8±6.0	91.4±5.1	82.3±9.5	88.2±6.5	82.1±8.5	85.9±6.6	83.7±8.4	86.5±6.8	90.7±4.4	86.0±4.6	89.7±7.5	91.2±4.9	83.6±8.2
	.20	84.1±7.6	85.9±7.6	78.7±10.5	85.3±7.9	77.4±10.9	82.6±8.6	78.6±10.3	81.5±8.4	87.4±5.6	82.1±5.4	87.4±8.6	86.6±6.6	79.7±9.4
F-MNIST	.00	92.8±4.7	91.2±4.7	91.6±5.5	82.5±8.1	92.0±4.9	69.7±14.4	90.2±5.8	89.2±6.2	94.0±4.4	90.5±5.9	71.4±12.7	90.5±6.5	76.8±13.2
	.01	91.7±5.0	91.5±4.6	91.5±5.5	84.9±7.2	89.4±6.3	73.9±12.4	87.1±7.3	86.3±6.3	92.2±4.9	87.8±6.1	71.2±14.3	87.2±7.1	67.3±8.1
	.05	90.7±5.5	90.7±4.9	90.9±5.9	85.5±7.2	85.2±9.1	75.4±12.9	81.6±9.6	80.6±7.1	88.3±6.2	82.7±7.8	71.9±14.3	81.5±8.5	59.8±4.6
	.10	89.5±6.1	89.3±6.2	90.2±6.3	85.5±7.7	81.8±11.2	77.8±12.0	77.4±11.1	76.2±7.3	85.6±7.0	79.8±9.0	72.5±15.5	78.2±9.1	56.7±4.1
	.20	86.3±7.7	88.1±6.9	88.4±7.6	86.3±7.4	77.4±13.6	82.1±9.8	72.5±12.6	69.3±6.3	81.9±8.1	74.3±10.6	70.8±16.0	74.8±9.4	53.9±2.9
CIFAR-10	.00	62.0±10.6	63.8±9.0	60.0±10.0	59.9±6.7	59.9±11.7	56.1±10.2	56.2±13.2	60.9±9.4	73.8±7.6	73.3±8.4	50.8±4.7	77.9±7.2	63.5±8.0
	.01	61.9±10.6	63.8±9.3	59.9±10.1	59.9±6.7	59.2±12.3	56.3±10.4	56.2±13.1	60.5±9.4	73.0±8.0	72.8±8.1	51.1±4.7	76.5±7.2	62.9±7.3
	.05	61.4±10.7	62.6±9.2	59.6±10.1	59.6±6.4	58.1±12.9	55.6±10.5	55.7±13.3	59.6±9.8	71.5±8.2	71.0±8.4	50.1±2.9	74.0±6.9	62.2±8.2
	.10	60.8±10.7	62.9±8.2	58.8±10.1	59.1±6.6	57.3±13.5	54.9±11.1	55.4±13.3	58.6±10.0	69.8±8.4	69.3±8.5	50.5±3.6	71.8±7.0	60.6±8.3
	.20	60.3±10.3	61.9±8.1	57.9±10.1	58.3±6.2	56.2±13.9	54.2±11.1	54.6±13.3	57.0±10.6	67.8±8.6	67.9±8.1	50.1±1.7	68.5±7.1	58.5±6.7

Table C.4: Complete results of experimental scenario (iii), where we increase the number of anomaly classes k_l included in the labeled training data. We report the mean AUC with std. dev. computed over 100 experiments at various numbers k_l.

	k_l	OC-SVM Raw	OC-SVM Hybrid	iForest Raw	iForest Hybrid	KDE Raw	KDE Hybrid	AE	Deep SVDD	SSAD Raw	SSAD Hybrid	SS-DGM	Deep SAD	Supervised Classifier
MNIST	0	88.8±6.0	91.4±5.1	82.3±9.5	88.2±6.5	82.1±8.5	85.9±6.6	83.7±8.4	86.5±6.8	88.8±6.0	91.4±5.1		86.5±6.8	
	1									90.7±4.4	86.0±4.6	89.7±7.5	91.2±4.9	83.6±8.2
	2									92.5±3.6	87.7±3.8	92.8±5.3	92.0±3.6	90.3±4.6
	3									93.9±3.3	89.8±3.3	94.9±4.2	94.7±2.8	93.9±2.8
	5									95.5±2.5	91.9±3.0	96.7±2.3	97.3±1.8	96.9±1.7
F-MNIST	0	89.5±6.1	89.3±6.2	90.2±6.3	85.5±7.7	81.8±11.2	77.8±12.0	77.4±11.1	76.2±7.3	89.5±6.1	89.3±6.2		76.2±7.3	
	1									85.6±7.0	79.8±9.0	72.5±15.5	78.2±9.1	56.7±4.1
	2									87.8±6.1	80.1±10.5	74.3±15.4	80.5±8.2	62.3±2.9
	3									89.4±5.5	83.8±9.4	77.5±14.7	83.9±7.4	67.3±3.0
	5									91.2±4.8	86.8±7.7	79.9±13.8	87.3±6.4	75.3±2.7
CIFAR-10	0	60.8±10.7	62.9±8.2	58.8±10.1	59.1±6.6	57.3±13.5	54.9±11.1	55.4±13.3	58.6±10.0	60.8±10.7	62.9±8.2		58.6±10.0	
	1									69.8±8.4	69.3±8.5	50.5±3.6	71.8±7.0	60.6±8.3
	2									73.0±7.1	72.3±7.5	50.3±2.4	75.2±6.4	61.0±6.6
	3									73.8±6.6	73.3±7.0	50.0±0.7	77.5±5.9	62.7±6.8
	5									75.1±5.5	74.2±6.5	50.0±1.0	80.4±4.6	60.9±4.6

C.3 FCDD Results on Individual Classes

Table C.5 reports the class-wise results on Fashion-MNIST for AE, Deep SVDD (DSVDD), and Geometric Transformations (GT) [181].

Table C.5: Detection performance in mean AUC (over 5 seeds) for all classes of Fashion-MNIST [589].

	without OE			with OE
	AE	DSVDD	GT*	FCDD
t-shirt/top	0.85	0.98	0.99	0.82
trouser	0.91	0.90	0.98	0.98
pullover	0.78	0.91	0.91	0.84
dress	0.88	0.94	0.90	0.92
coat	0.88	0.89	0.92	0.87
sandal	0.45	0.92	0.93	0.90
shirt	0.70	0.83	0.83	0.75
sneaker	0.96	0.99	0.99	0.99
bag	0.87	0.92	0.91	0.86
ankle boot	0.96	0.99	0.99	0.94
mean	0.82	0.93	0.94	0.89

In Table C.6, we report the class-wise results for CIFAR-10. Competitors without OE are AE, DSVDD, GT [181] and an adaptation of GT (GT+) [222]. Competitors with OE are the focal loss classifier [222], again GT+ [222], Deep SAD, and HSC.

Table C.6: Detection performance in mean AUC (over 5 seeds) for all classes of CIFAR-10 [290].

	without OE				with OE				
	AE	DSVDD	GT*	GT+*	GT+*	Focal*	DSAD	HSC	FCDD
airplane	0.59	0.62	0.75	0.78	0.90	0.88	0.94	0.97	0.95
automobile	0.57	0.66	0.96	0.97	0.99	0.94	0.98	0.99	0.96
bird	0.49	0.51	0.78	0.87	0.94	0.79	0.90	0.93	0.91
cat	0.58	0.59	0.72	0.81	0.88	0.80	0.87	0.90	0.90
deer	0.54	0.61	0.88	0.93	0.97	0.82	0.95	0.97	0.94
dog	0.62	0.66	0.88	0.90	0.94	0.86	0.93	0.94	0.93
frog	0.51	0.68	0.83	0.91	0.97	0.93	0.97	0.98	0.97
horse	0.59	0.67	0.96	0.97	0.99	0.88	0.97	0.98	0.96
ship	0.77	0.76	0.93	0.95	0.99	0.93	0.97	0.98	0.97
truck	0.67	0.73	0.91	0.93	0.99	0.92	0.96	0.97	0.96
mean	0.59	0.65	0.86	0.90	0.96	0.87	0.95	0.96	0.95

In Table C.7, we report the class-wise results for ImageNet, where competitors are the AE, the focal loss classifier [222], GT+ [222], Deep SAD, and HSC. Results from the literature are marked with an asterisk.

Table C.7: Detection performance in mean AUC (over 10 seeds) for 30 classes of ImageNet [133]. The results from the literature, Focal* and GT+*, do not report results per class.

	without OE	with OE				
	AE	Focal*	GT+*	DSAD	HSC	FCDD
acorn	0.45	×	×	0.99	0.99	0.97
airliner	0.80	×	×	0.97	1.00	0.98
ambulance	0.25	×	×	0.99	1.00	0.99
american alligator	0.61	×	×	0.93	0.98	0.97
banjo	0.45	×	×	0.97	0.98	0.91
barn	0.59	×	×	0.99	1.00	0.97
bikini	0.46	×	×	0.97	0.99	0.94
digital clock	0.63	×	×	0.99	0.97	0.92
dragonfly	0.62	×	×	0.99	0.98	0.98
dumbbell	0.42	×	×	0.93	0.92	0.88
forklift	0.28	×	×	0.91	0.99	0.94
goblet	0.63	×	×	0.92	0.94	0.90
grand piano	0.45	×	×	1.00	0.97	0.95
hotdog	0.48	×	×	0.96	0.99	0.97
hourglass	0.58	×	×	0.96	0.97	0.92
manhole cover	0.70	×	×	0.99	1.00	1.00
mosque	0.72	×	×	0.99	0.99	0.97
nail	0.57	×	×	0.93	0.94	0.92
parking meter	0.45	×	×	0.99	0.93	0.87
pillow	0.40	×	×	0.99	0.94	0.94
revolver	0.60	×	×	0.98	0.98	0.93
rotary dial telephone	0.58	×	×	0.90	0.98	0.91
schooner	0.65	×	×	0.99	0.99	0.96
snowmobile	0.54	×	×	0.98	0.99	0.97
soccer ball	0.46	×	×	0.97	0.93	0.86
stingray	0.84	×	×	0.99	0.99	0.97
strawberry	0.44	×	×	0.98	0.99	0.97
tank	0.57	×	×	0.97	0.99	0.96
toaster	0.59	×	×	0.98	0.92	0.79
volcano	0.90	×	×	0.90	1.00	0.97
mean	0.56	0.56	0.86	0.97	0.97	0.94

C.4 Average Precision on MNIST-C and MVTec-AD

We provide the detection performance measured in Average Precision (AP) of the experimental evaluation on MNIST-C and MVTec-AD from Section 4.4.1 in Tables C.8 and C.9 respectively. As can be seen (and as to be expected [129]), the performance in AP shows the same trends as AUC (see Tables 4.2 and 4.3 in Section 4.4.1), since the MNIST-C and MVTec-AD test sets are not highly imbalanced.

Table C.8: Mean AP (in %) detection performance (over 5 seeds) on MNIST-C.

	Gaussian	MVE	PCA	KDE	SVDD	kPCA	AGAN	DSVDD	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 C.9: Mean AP (in %) detection performance (over 5 seeds) on MVTec-AD.

		Gaussian	MVE	PCA	KDE	SVDD	kPCA	AGAN	DSVDD	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

Bibliography

[1] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara. Latent space autoregression for novelty detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2019.

[2] A. Abdallah, M. A. Maarof, and A. Zainal. Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68:90–113, 2016.

[3] A. Abdelhamed, M. A. Brubaker, and M. S. Brown. Noise flow: Noise modeling with conditional normalizing flows. In *International Conference on Computer Vision*, pages 3165–3173, 2019.

[4] N. Abe, B. Zadrozny, and J. Langford. Outlier detection by active learning. In *International Conference on Knowledge Discovery & Data Mining*, pages 504–509, 2006.

[5] A. Achille and S. Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

[6] C. C. Aggarwal. *Outlier Analysis*. Springer International Publishing, 2nd edition, 2017.

[7] S. Agrawal and J. Agrawal. Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, 60:708–713, 2015.

[8] H. Aguinis, R. K. Gottfredson, and H. Joo. Best-practice recommendations for defining, identifying, and handling outliers. *Organizational Research Methods*, 16(2):270–301, 2013.

[9] F. Ahmed and A. Courville. Detecting semantic anomalies. In *AAAI Conference on Artificial Intelligence*, pages 3154–3162, 2020.

[10] M. Ahmed. Collective anomaly detection techniques for network traffic analysis. *Annals of Data Science*, 5(4):497–512, 2018.

[11] M. Ahmed, A. N. Mahmood, and J. Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.

[12] M. Ahmed, A. N. Mahmood, and M. R. Islam. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278–288, 2016.

[13] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon. GANomaly: Semi-supervised anomaly detection via adversarial training. In *Asian Conference on Computer Vision*, pages 622–637, 2018.

[14] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: A survey. *Data Mining and Knowledge Discovery*, 29(3):626–688, 2015.

[15] A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, and K. Murphy. Fixing a broken ELBO. In *International Conference on Machine Learning*, volume 80, pages 159–168, 2018.

[16] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

[17] T. Amarbayasgalan, B. Jargalsaikhan, and K. H. Ryu. Unsupervised novelty detection using deep autoencoders with density based clustering. *Applied Sciences*, 8(9):1468, 2018.

[18] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Elsen, J. Engel, W. Fang, L. Fan, C. Fougner, L. Gao, C. Gong, A. Hannun, T. Han, L. Johannes, B. Jiang, C. Ju, B. Jun, P. LeGresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wu, L. Wei, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan, and Z. Zhu. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, volume 48, pages 173–182, 2016.

[19] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[20] N. Amruthnath and T. Gupta. A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In *International Conference on Industrial Engineering and Applications (ICIEA)*, pages 355–361. IEEE, 2018.

[21] J. An and S. Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2:1–18, 2015.

[22] C. Anders, P. Pasliev, A.-K. Dombrowski, K.-R. Müller, and P. Kessel. Fairwashing explanations with off-manifold detergent. In *International Conference on Machine Learning*, volume 119, pages 314–323, 2020.

[23] F. J. Anscombe. Rejection of outliers. *Technometrics*, 2(2):123–146, May 1960.

[24] F. Arcadu, F. Benmansour, A. Maunz, J. Willis, Z. Haskova, and M. Prunotto. Deep learning algorithm predicts diabetic retinopathy progression in individual patients. *npj Digital Medicine*, 2(1):1–9, 2019.

[25] D. Ardila, A. P. Kiraly, S. Bharadwaj, B. Choi, J. J. Reicher, L. Peng, D. Tse, M. Etemadi, W. Ye, G. Corrado, et al. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature Medicine*, 25(6):954–961, 2019.

[26] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, volume 70, pages 214–223, 2017.

[27] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

[28] S. Arora, Y. Liang, and T. Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*, 2017.

[29] S. Arora, N. Cohen, and E. Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, volume 80, pages 244–253, 2018.

[30] D. Arpit, Y. Zhou, H. Ngo, and V. Govindaraju. Why regularized auto-encoders learn sparse representation? In *International Conference on Machine Learning*, volume 48, pages 136–144, 2016.

[31] L. Arras, F. Horn, G. Montavon, K.-R. Müller, and W. Samek. "what is relevant in a text document?": An interpretable machine learning approach. *PLOS ONE*, 12(8):e0181142, 2017.

[32] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, 2007.

[33] Y. M. Asano, C. Rupprecht, and A. Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *International Conference on Learning Representations*, 2020.

[34] D. J. Atha and M. R. Jahanshahi. Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Structural Health Monitoring*, 17(5): 1110–1128, 2018.

[35] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, volume 80, pages 274–283, 2018.

[36] C. Aytekin, X. Ni, F. Cricri, and E. Aksu. Clustering and unsupervised anomaly detection with l_2 normalized deep auto-encoder representations. In *International Joint Conference on Neural Networks*, pages 1–6, 2018.

[37] 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*, 10(7):e0130140, 2015.

[38] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun): 1803–1831, 2010.

[39] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.

[40] P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5:4308, 2014.

[41] D. H. Ballard. Modular learning in neural networks. In *AAAI Conference on Artificial Intelligence*, pages 279–284, 1987.

[42] V. Barnett and T. Lewis. *Outliers in Statistical Data*. Wiley, 3rd edition, 1994.

[43] P. L. Bartlett and M. H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(Aug):1823–1840, 2008.

[44] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab. Fusing unsupervised and supervised deep learning for white matter lesion segmentation. In *Medical Imaging with Deep Learning*, pages 63–72, 2019.

[45] M. Belkin, S. Ma, and S. Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pages 540–548, 2018.

[46] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[47] S. Ben-David and M. Lindenbaum. Learning distributions by their density levels: A paradigm for learning without a teacher. *Journal of Computer and System Sciences*, 55(1):171–182, 1997.

[48] A. Bendale and T. E. Boult. Towards open set deep networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1563–1572, 2016.

[49] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003.

[50] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8): 1798–1828, 2013.

[51] T. Berger. Rate-distortion theory. *Wiley Encyclopedia of Telecommunications*, 2003.

[52] L. Bergman and Y. Hoshen. Classification-based anomaly detection for general data. In *International Conference on Learning Representations*, 2020.

[53] L. Bergman, N. Cohen, and Y. Hoshen. Deep nearest neighbor anomaly detection. *arXiv preprint arXiv:2002.10445*, 2020.

[54] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. MVTec AD–A comprehensive real-world dataset for unsupervised anomaly detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9592–9600, 2019.

[55] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, pages 908–918, 2017.

[56] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland. Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3):602–613, 2011.

[57] B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

[58] A. Binder, M. Bockmayr, M. Hägele, S. Wienert, D. Heim, K. Hellweg, M. Ishii, A. Stenzinger, A. Hocke, C. Denkert, K.-R. Müller, and F. Klauschen. Morphological and molecular breast cancer profiling through explainable machine learning. *Nature Machine Intelligence*, pages 1–12, 2021.

[59] S. Bird, E. Loper, and E. Klein. *Natural Language Processing with Python*. O'Reilly Media, Inc., 2009.

[60] C. M. Bishop. Novelty detection and neural network validation. *IEE Proceedings - Vision, Image and Signal Processing*, 141(4):217–222, 1994.

[61] C. M. Bishop. Bayesian PCA. In *Advances in Neural Information Processing Systems*, pages 382–388, 1999.

[62] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer New York, 2006.

[63] G. Blanchard, G. Lee, and C. Scott. Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11(Nov):2973–3009, 2010.

[64] R. Blender, K. Fraedrich, and F. Lunkeit. Identification of cyclone-track regimes in the north atlantic. *Quarterly Journal of the Royal Meteorological Society*, 123(539):727–741, 1997.

[65] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. In *International Conference on Machine Learning*, volume 37, pages 1613–1622, 2015.

[66] P. Bojanowski and A. Joulin. Unsupervised learning by predicting noise. In *International Conference on Machine Learning*, volume 70, pages 517–526, 2017.

[67] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[68] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann. NetGAN: Generating graphs via random walks. In *International Conference on Machine Learning*, volume 80, pages 610–619, 2018.

[69] R. J. Bolton and D. J. Hand. Statistical fraud detection: A review. *Statistical Science*, 17(3): 235–255, 2002.

[70] L. Bontemps, J. McDermott, N.-A. Le-Khac, et al. Collective anomaly detection based on long short-term memory recurrent neural networks. In *International Conference on Future Data and Security Engineering*, pages 141–152. Springer, 2016.

[71] G. Boracchi, D. Carrera, C. Cervellera, and D. Maccio. QuantTree: Histograms for change detection in multivariate data streams. In *International Conference on Machine Learning*, volume 80, pages 639–648, 2018.

[72] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini. Anomaly detection using autoencoders in high performance computing systems. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 9428–9433, 2019.

[73] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. In *The SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Aug. 2016.

[74] G. E. Box. Science and statistics. *Journal of the American Statistical Association*, 71(356): 791–799, 1976.

[75] K. Boyd, K. H. Eng, and C. D. Page. Area under the precision-recall curve: Point estimates and confidence intervals. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 451–466, 2013.

[76] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[77] M. L. Braun, J. M. Buhmann, and K.-R. Müller. On relevant dimensions in kernel feature spaces. *Journal of Machine Learning Research*, 9(Aug):1875–1908, 2008.

[78] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 93–104, 2000.

[79] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

[80] 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. *arXiv preprint arXiv:2006.09000*, 2020.

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

[82] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.

[83] V. L. Cao, M. Nicolau, and J. McDermott. A hybrid autoencoder and density estimation model for anomaly detection. In *International Conference on Parallel Problem Solving from Nature*, pages 717–726. Springer International Publishing, 2016.

[84] G. Carleo and M. Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[85] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017.

[86] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[87] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, pages 132–149, 2018.

[88] 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 *International Conference on Knowledge Discovery & Data Mining*, pages 1721–1730, 2015.

[89] O. Cerri, T. Q. Nguyen, M. Pierini, M. Spiropulu, and J.-R. Vlimant. Variational autoencoders for new physics mining at the large hadron collider. *Journal of High Energy Physics*, 2019(5): 36, 2019.

[90] R. Chalapathy and S. Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[91] R. Chalapathy, A. K. Menon, and S. Chawla. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 36–51, 2017.

[92] R. Chalapathy, A. K. Menon, and S. Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

[93] R. Chalapathy, E. Toth, and S. Chawla. Group anomaly detection using deep generative models. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 173–189, 2018.

[94] W. Chan, N. Jaitly, Q. Le, and O. Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 4960–4964, 2016.

[95] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009.

[96] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):823–839, 2010.

[97] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, Cambridge, Massachusetts, 2006.

[98] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997.

[99] S. Chauhan and L. Vig. Anomaly detection in ECG time signals via deep long short-term memory networks. In *IEEE International Conference on Data Science and Advanced Analytics*, pages 1–7, 2015.

[100] S. Chawla and P. Sun. SLOM: a new measure for local spatial outliers. *Knowledge and Information Systems*, 9(4):412–429, 2006.

[101] T. Che, X. Liu, S. Li, Y. Ge, R. Zhang, C. Xiong, and Y. Bengio. Deep verifier networks: Verification of deep discriminative models with deep generative models. *arXiv preprint arXiv:1911.07421*, 2019.

[102] P. Cheema, N. L. D. Khoa, M. Makki Alamdari, W. Liu, Y. Wang, F. Chen, and P. Runcie. On structural health monitoring using tensor analysis and support vector machine with artificial negative data. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1813–1822, 2016.

[103] J. Chen, S. Sathe, C. C. Aggarwal, and D. S. Turaga. Outlier detection with autoencoder ensembles. In *SIAM International Conference on Data Mining*, pages 90–98, 2017.

[104] L. Chen, S. Dai, C. Tao, H. Zhang, Z. Gan, D. Shen, Y. Zhang, G. Wang, R. Zhang, and L. Carin. Adversarial text generation via feature-mover's distance. In *Advances in Neural Information Processing Systems*, pages 4666–4677, 2018.

[105] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 10709–10719, 2020.

[106] X. Chen and E. Konukoglu. Unsupervised detection of lesions in brain MRI using constrained adversarial auto-encoders. In *Medical Imaging with Deep Learning*, 2018.

[107] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

[108] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014.

[109] H. Choi, E. Jang, and A. A. Alemi. WAIC, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

[110] S. Choi and S.-Y. Chung. Novelty detection via blurring. In *International Conference on Learning Representations*, 2020.

[111] P. Chong, L. Ruff, M. Kloft, and A. Binder. Simple and effective prevention of mode collapse in deep one-class classification. In *International Joint Conference on Neural Networks*, pages 1–9, 2020.

[112] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord. Unsupervised speech representation learning using WaveNet autoencoders. *IEEE Transactions on Audio, Speech, and Language Processing*, 27(12):2041–2053, 2019.

[113] C. K. Chow. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, EC-6(4):247–254, Dec. 1957.

[114] C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

[115] S. Clémençon and J. Jakubowicz. Scoring anomalies: a m-estimation formulation. In *International Conference on Artificial Intelligence and Statistics*, pages 659–667, 2013.

[116] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *International Joint Conference on Neural Networks*, pages 2921–2926, 2017.

[117] N. Cohen, O. Sharir, and A. Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, volume 49, pages 698–728, 2016.

[118] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning*, pages 160–167, 2008.

[119] C. Cortes, G. DeSalvo, and M. Mohri. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, pages 67–82, 2016.

[120] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006.

[121] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2011.

[122] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov. Good semi-supervised learning that requires a bad GAN. In *Advances in Neural Information Processing Systems*, volume 30, pages 6510–6520, 2017.

[123] X. H. Dang, B. Micenková, I. Assent, and R. T. Ng. Local outlier detection with interpretation. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 304–320, 2013.

[124] X. H. Dang, I. Assent, R. T. Ng, A. Zimek, and E. Schubert. Discriminative features for identifying and interpreting outliers. In *International Conference on Data Engineering*, pages 88–99. IEEE, 2014.

[125] T. Daniel, T. Kurutach, and A. Tamar. Deep variational semi-supervised novelty detection. *arXiv preprint arXiv:1911.04971*, 2019.

[126] 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 *International Conference on Knowledge Discovery & Data Mining*, pages 47–56, 2010.

[127] S. Das, W.-K. Wong, T. Dietterich, A. Fern, and A. Emmott. Discovering anomalies by incorporating feedback from an expert. *Transactions on Knowledge Discovery from Data*, 14 (4):1–32, 2020.

[128] M. A. Davenport, R. G. Baraniuk, and C. D. Scott. Learning minimum volume sets with support vector machines. In *IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, pages 301–306, 2006.

[129] J. Davis and M. Goadrich. The relationship between precision-recall and ROC curves. In *International Conference on Machine Learning*, pages 233–240, 2006.

[130] L. Deecke, R. A. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft. Image anomaly detection with generative adversarial networks. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 3–17, 2018.

[131] L. Deecke, L. Ruff, R. A. Vandermeulen, and H. Bilen. Transfer-based semantic anomaly detection. In *International Conference on Machine Learning*, volume 139, pages 2546–2558, 2021.

[132] D. Dehaene, O. Frigo, S. Combrexelle, and P. Eline. Iterative energy-based projection on a normal data manifold for anomaly localization. In *International Conference on Learning Representations*, 2020.

[133] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[134] F. Denis. PAC learning from positive statistical queries. In *International Conference on Algorithmic Learning Theory*, pages 112–126, 1998.

[135] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019.

[136] T. DeVries and G. W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

[137] L. Devroye and L. Györfi. *Nonparametric Density Estimation: The L1 View*. John Wiley & Sons, New York; Chichester, 1985.

[138] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means, spectral clustering and normalized cuts. In *International Conference on Knowledge Discovery & Data Mining*, pages 551–556, 2004.

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

[140] L. Dinh, D. Krueger, and Y. Bengio. NICE: non-linear independent components estimation. In *International Conference on Learning Representations*, 2015.

[141] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017.

[142] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision*, pages 1422–1430, 2015.

[143] 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 *Advances in Neural Information Processing Systems*, pages 13589–13600, 2019.

[144] M. Du, Z. Chen, C. Liu, R. Oak, and D. Song. Lifelong anomaly detection through unlearning. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 1283–1297, 2019.

[145] M. C. Du Plessis, G. Niu, and M. Sugiyama. Analysis of learning from positive and unlabeled data. In *Advances in Neural Information Processing Systems*, pages 703–711, 2014.

[146] D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[147] L. Duan, G. Tang, J. Pei, J. Bailey, A. Campbell, and C. Tang. Mining outlying aspects on numeric data. *Data Mining and Knowledge Discovery*, 29(5):1116–1151, 2015.

[148] F. Dufrenois. A one-class kernel fisher criterion for outlier detection. *IEEE Transactions on Neural Networks and Learning Systems*, 26(5):982–994, 2014.

[149] H. Dutta, C. Giannella, K. Borne, and H. Kargupta. Distributed top-k outlier detection from astronomy catalogs using the DEMAC system. In *SIAM International Conference on Data Mining*, pages 473–478, 2007.

[150] F. Y. Edgeworth. On discordant observations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 23(5):364–375, 1887.

[151] R. El-Yaniv and M. Nisenson. Optimal single-class classification strategies. In *Advances in Neural Information Processing Systems*, pages 377–384, 2007.

[152] R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *Conference on Learning Theory*, volume 49, pages 907–940, 2016.

[153] A. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong. A meta-analysis of the anomaly detection problem. *arXiv preprint arXiv:1503.01158*, 2016.

[154] S. Erfani, M. Baktashmotlagh, S. Rajasegarar, S. Karunasekera, and C. Leckie. R1SVM: A randomised nonlinear approach to large-acale anomaly detection. In *AAAI Conference on Artificial Intelligence*, pages 432–438, 2015.

[155] 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 Recognition*, 58: 121–134, 2016.

[156] T. Ergen and S. S. Kozat. Unsupervised anomaly detection with LSTM neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 31(8):3127–3141, 2020.

[157] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29, 2019.

[158] S. E. Fahlman, G. E. Hinton, and T. J. Sejnowski. Massively parallel architectures for AI: NETL, Thistle, and Boltzmann machines. In *AAAI Conference on Artificial Intelligence*, pages 109–113, 1983.

[159] W. Fan, M. Miller, S. Stolfo, W. Lee, and P. Chan. Using artificial anomalies to detect unknown and known network intrusions. *Knowledge and Information Systems*, 6(5):507–527, 2004.

[160] K. Faust, Q. Xie, D. Han, K. Goyle, Z. Volynskaya, U. Djuric, and P. Diamandis. Visualizing histopathologic deep learning classification and anomaly detection using nonlinear feature space dimensionality reduction. *BMC Bioinformatics*, 19(1):173, 2018.

[161] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[162] 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. *Journal of Computational Science*, 20:143–153, 2017.

[163] M. Flach, F. Gans, A. Brenning, J. Denzler, M. Reichstein, E. Rodner, S. Bathiany, P. Bodesheim, Y. Guanche, S. Sippel, et al. Multivariate anomaly detection for earth observations: a comparison of algorithms and feature extraction techniques. *Earth System Dynamics*, 8(3):677–696, 2017.

[164] E. Fouché, Y. Meng, F. Guo, H. Zhuang, K. Böhm, and J. Han. Mining text outliers in document directories. In *IEEE International Conference on Data Mining*, pages 152–161, 2020.

[165] A. J. Fox. Outliers in time series. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(3):350–363, 1972.

[166] S. Frühwirth-Schnatter. *Finite Mixture and Markov Switching Models*. Springer Science & Business Media, 2006.

[167] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, volume 48, pages 1050–1059, 2016.

[168] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[169] C. Gautam, R. Balaji, K. Sudharsan, A. Tiwari, and K. Ahuja. Localized multiple kernel learning for anomaly detection: One-class classification. *Knowledge-Based Systems*, 165: 241–252, 2019.

[170] Y. Geifman and R. El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 4878–4887, 2017.

[171] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*, volume 159 of *The Springer International Series in Engineering and Computer Science*. Springer, Boston, MA, 1992.

[172] Z. Ghafoori and C. Leckie. Deep multi-sphere support vector data description. In *SIAM International Conference on Data Mining*, pages 109–117, 2020.

[173] L. E. Ghaoui, M. I. Jordan, and G. R. Lanckriet. Robust novelty detection with single-class MPM. In *Advances in Neural Information Processing Systems*, pages 929–936, 2003.

[174] A. Ghasemi, H. R. Rabiee, M. T. Manzuri, and M. H. Rohban. A bayesian approach to the data description problem. In *AAAI Conference on Artificial Intelligence*, pages 907–913, 2012.

[175] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[176] A. Glazer, M. Lindenbaum, and S. Markovitch. q-OCSVM: A q-quantile estimator for high-dimensional distributions. In *Advances in Neural Information Processing Systems*, pages 503–511, 2013.

[177] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, volume 9, pages 249–256, 2010.

[178] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita. A survey of outlier detection methods in network anomaly identification. *Computer Journal*, 54(4):570–588, 2011.

[179] G. Goh. Why momentum really works. *Distill*, 2017. doi: 10.23915/distill.00006. URL `http://distill.pub/2017/momentum`.

[180] N. Goix, A. Sabourin, and S. Clémençon. On anomaly ranking and excess-mass curves. In *International Conference on Artificial Intelligence and Statistics*, pages 287–295, 2015.

[181] I. Golan and R. El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, pages 9758–9769, 2018.

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

[183] K. Golmohammadi and O. R. Zaiane. Time series contextual anomaly detection for detecting market manipulation in stock market. In *IEEE International Conference on Data Science and Advanced Analytics*, pages 1–10, 2015.

[184] K. Golmohammadi and O. R. Zaiane. Sentiment analysis on twitter to improve time series contextual anomaly detection for detecting stock market manipulation. In *International Conference on Big Data Analytics and Knowledge Discovery*, pages 327–342, 2017.

[185] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *International Conference on Computer Vision*, pages 1705–1714, 2019.

[186] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

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

[188] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[189] B. Goodman and S. Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, 2017.

[190] N. Görnitz, M. Kloft, and U. Brefeld. Active and semi-supervised data domain description. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 407–422, 2009.

[191] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 46:235–262, 2013.

[192] N. Görnitz, L. A. Lima, K.-R. Müller, M. Kloft, and S. Nakajima. Support vector data descriptions and k-means clustering: One class? *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):3994–4006, 2017.

[193] S. Goyal, A. Raghunathan, M. Jain, H. V. Simhadri, and P. Jain. DROCC: Deep robust one-class classification. In *International Conference on Machine Learning*, pages 11335–11345, 2020.

[194] Y. Grandvalet, A. Rakotomamonjy, J. Keshet, and S. Canu. Support vector machines with a reject option. In *Advances in Neural Information Processing Systems*, pages 537–544, 2009.

[195] 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 *International Conference on Learning Representations*, 2020.

[196] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 6645–6649, 2013.

[197] P. S. Gromski, A. B. Henson, J. M. Granda, and L. Cronin. How to explore chemical space using algorithms and automation. *Nature Reviews Chemistry*, 3(2):119–128, 2019.

[198] F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11 (1):1–21, 1969.

[199] X. Gu, L. Akoglu, and A. Rinaldo. Statistical analysis of nearest neighbor methods for anomaly detection. In *Advances in Neural Information Processing Systems*, pages 10923–10933, 2019.

[200] S. Guha, N. Mishra, G. Roy, and O. Schrijvers. Robust random cut forest based anomaly detection on streams. In *International Conference on Machine Learning*, volume 48, pages 2712–2721, 2016.

[201] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[202] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, volume 70, pages 1321–1330, 2017.

[203] P. Guo, Z. Xue, Z. Mtema, K. Yeates, O. Ginsburg, M. Demarco, L. R. Long, M. Schiffman, and S. Antani. Ensemble deep learning for cervix image selection toward improving reliability in automated cervical precancer screening. *Diagnostics*, 10(7):451, 2020.

[204] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

[205] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. Outlier detection for temporal data: A survey. *Transactions on Knowledge and Data Engineering*, 26(9):2250–2267, 2014.

[206] D. Guthrie. *Unsupervised Detection of Anomalous Text*. PhD thesis, University of Sheffield, 2008.

[207] M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

[208] A. S. Hadi, R. Imon, and M. Werner. Detection of outliers. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1):57–70, 2009.

[209] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *International Conference on Machine Learning*, page 47, 2004.

[210] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons, 2005.

[211] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[212] W. Härdle. *Applied Nonparametric Regression*. Econometric Society Monographs. Cambridge university press, 1990.

[213] S. Harmeling, G. Dornhege, D. M. J. Tax, F. Meinecke, and K.-R. Müller. From outliers to prototypes: ordering data. *Neurocomputing*, 69(13-15):1608–1618, 2006.

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

[215] D. M. Hawkins. The detection of errors in multivariate data using principal components. *Journal of the American Statistical Association*, 69(346):340–344, 1974.

[216] D. M. Hawkins. *Identification of Outliers*, volume 11. Springer, 1980.

[217] S. Hawkins, H. He, G. Williams, and R. Baxter. Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, volume 2454, pages 170–180, 2002.

[218] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[219] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

[220] D. Hendrycks, K. Lee, and M. Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, volume 97, pages 2712–2721, 2019.

[221] D. Hendrycks, M. Mazeika, and T. G. Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

[222] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, pages 15637–15648, 2019.

[223] 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*, pages 149–184. Springer, 2013.

[224] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori. Statistical outlier detection using direct density ratio estimation. *Knowledge and Information Systems*, 26(2):309–336, 2011.

[225] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[226] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[227] G. E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40(1):185–234, 1989.

[228] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, Aug. 2002.

[229] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, Jul. 2006.

[230] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, July 2006.

[231] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

[232] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, Oct. 2004.

[233] C. D. Hofer, R. Kwitt, M. Dixit, and M. Niethammer. Connectivity-optimized representation learning via persistent homology. In *International Conference on Machine Learning*, volume 97, pages 2751–2760, 2019.

[234] H. Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007.

[235] J. Höner, S. Nakajima, A. Bauer, K.-R. Müller, and N. Görnitz. Minimizing trust leaks for robust sybil detection. In *International Conference on Machine Learning*, volume 70, pages 1520–1528, 2017.

[236] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79 (8):2554–2558, 1982.

[237] A. Høst-Madsen, E. Sabeti, and C. Walton. Data discovery and anomaly detection using atypicality: Theory. *Transactions on Information Theory*, 65(9):5302–5322, 2019.

[238] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[239] C. Huang, J. Cao, F. Ye, M. Li, Y. Zhang, and C. Lu. Inverse-transform autoencoder for anomaly detection. *arXiv preprint arXiv:1911.10676*, 2019.

[240] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, volume 80, pages 2078–2087, 2018.

[241] F. J. Huang and Y. LeCun. Large-scale learning with SVM and convolutional nets for generic object categorization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–291, 2006.

[242] L. Huang, X. Nguyen, M. Garofalakis, M. I. Jordan, A. Joseph, and N. Taft. In-network PCA and anomaly detection. In *Advances in Neural Information Processing Systems*, pages 617–624, 2007.

[243] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[244] P. J. Huber and E. M. Ronchetti. *Robust Statistics*. John Wiley & Sons, 2nd edition, 2009.

[245] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *International Conference on Knowledge Discovery & Data Mining*, pages 387–395, 2018.

[246] 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 Transactions on Medical Imaging*, 37(10):2196–2210, 2018.

[247] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

[248] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*, volume 37, pages 448–456, 2015.

[249] J. E. Jackson and G. S. Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21(3):341–349, Aug. 1979.

[250] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8): 651–666, 2010.

[251] A. K. Jain and R. C. Dubes. *aa*. Prentice-Hall, Inc., USA, 1988.

[252] N. Japkowicz, C. Myers, and M. Gluck. A novelty detection approach to classification. In *International Joint Conferences on Artificial Intelligence*, volume 1, pages 518–523, 1995.

[253] M. H. Jarrahi. Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making. *Business Horizons*, 61(4):577–586, 2018.

[254] X. Ji, J. F. Henriques, and A. Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *International Conference on Computer Vision*, pages 9865–9874, 2019.

[255] H. Jiang, B. Kim, M. Guan, and M. Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems*, pages 5541–5552, 2018.

[256] H. Jiang, H. Wang, W. Hu, D. Kakde, and A. Chaudhuri. Fast incremental SVDD learning algorithm with the Gaussian kernel. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 3991–3998, 2019.

[257] T. Jiang, Y. Li, W. Xie, and Q. Du. Discriminative reconstruction constrained generative adversarial network for hyperspectral anomaly detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[258] W. Jin, R. Barzilay, and T. Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, volume 80, pages 2323–2332, 2018.

[259] I. T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, New York, NY, 2nd edition, 2002.

[260] H. Joudaki, A. Rashidian, B. Minaei-Bidgoli, M. Mahmoodi, B. Geraili, M. Nasiri, and M. Arab. Using data mining to detect health care fraud and abuse: A review of literature. *Global Journal of Health Science*, 7(1):194–202, 2015.

[261] A. Joulin, É. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 427–431, 2017.

[262] P. Jurmeister, M. Bockmayr, P. Seegerer, T. Bockmayr, D. Treue, G. Montavon, C. Vollbrecht, A. Arnold, D. Teichmann, K. Bressem, et al. Machine learning analysis of DNA methylation profiles distinguishes primary lung squamous cell carcinomas from head and neck metastases. *Science Translational Medicine*, 11(509), 2019.

[263] P. Juszczak, D. M. J. Tax, E. Pe, R. P. W. Duin, et al. Minimum spanning tree based one-class classifier. *Neurocomputing*, 72(7):1859–1869, 2009.

[264] M. Kampffmeyer, S. Løkse, F. M. Bianchi, L. Livi, A.-B. Salberg, and R. Jenssen. Deep divergence-based approach to clustering. *Neural Networks*, 113:91–101, 2019.

[265] R. Kannan, H. Woo, C. C. Aggarwal, and H. Park. Outlier detection for text data. In *SIAM International Conference on Data Mining*, pages 489–497, 2017.

[266] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[267] J. Kauffmann, M. Esders, G. Montavon, W. Samek, and K.-R. Müller. From clustering to cluster explanations via neural networks. *arXiv preprint arXiv:1906.07633*, 2019.

[268] J. Kauffmann, K.-R. Müller, and G. Montavon. Towards explaining anomalies: A deep Taylor decomposition of one-class models. *Pattern Recognition*, 101:107198, 2020.

[269] J. Kauffmann, L. Ruff, G. Montavon, and K.-R. Müller. The Clever Hans effect in anomaly detection. *arXiv preprint arXiv:2006.10609*, 2020.

[270] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5574–5584, 2017.

[271] S. S. Khan and M. G. Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374, 2014.

[272] Y. A. Kharkov, V. Sotskov, A. Karazeev, E. Kiktenko, and A. Fedorov. Revealing quantum chaos with machine learning. *Physical Review B*, 101(6):064406, 2020.

[273] T. Kieu, B. Yang, C. Guo, and C. S. Jensen. Outlier detection for time series with recurrent autoencoder ensembles. In *International Joint Conferences on Artificial Intelligence*, pages 2725–2732, 2019.

[274] J. Kim and C. D. Scott. Robust kernel density estimation. *Journal of Machine Learning Research*, 13(82):2529–2565, 2012.

[275] K. H. Kim, S. Shim, Y. Lim, J. Jeon, J. Choi, B. Kim, and A. S. Yoon. RaPP: Novelty detection with reconstruction along projection pathway. In *International Conference on Learning Representations*, 2020.

[276] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[277] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[278] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[279] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

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

[281] P. Kirichenko, P. Izmailov, and A. G. Wilson. Why normalizing flows fail to detect out-of-distribution data. In *Advances in Neural Information Processing Systems*, 2020.

[282] F. Klauschen, K.-R. Müller, A. Binder, M. Bockmayr, M. Hägele, P. Seegerer, S. Wienert, G. Pruneri, S. de Maria, S. Badve, et al. Scoring of tumor-infiltrating lymphocytes: From visual estimation to machine learning. *Seminars in Cancer Biology*, 52(2):151, 2018.

[283] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: algorithms and applications. *The VLDB Journal*, 8(3):237–253, 2000.

[284] I. Kobyzev, S. Prince, and M. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[285] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, Sep. 1990.

[286] T. Kohonen. *Self-Organizing Maps*. Springer, 3rd edition, 2001.

[287] 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 Transactions on Audio, Speech, and Language Processing*, 27(1):212–224, 2018.

[288] M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[289] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *International Conference on Knowledge Discovery & Data Mining*, pages 444–452, 2008.

[290] A. Krizhevsky and G. E. Hinton. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009.

[291] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[292] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, R. L. Priol, and A. Courville. Out-of-distribution generalization via risk extrapolation. *arXiv preprint arXiv:2003.00688*, 2020.

[293] T. S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago Press, Chicago, 1970.

[294] R. Kumaraswamy, A. Wazalwar, T. Khot, J. W. Shavlik, and S. Natarajan. Anomaly detection in text: The value of domain knowledge. In *International Florida Artificial Intelligence Research Society Conference*, pages 225–228, 2015.

[295] N. Kwak. Principal component analysis based on L1-norm maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1672–1680, 2008.

[296] 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 Computing*, 10:1–13, 2017.

[297] G. Kwon, M. Prabhushankar, D. Temel, and G. AlRegib. Backpropagated gradient representations for anomaly detection. In *European Conference on Computer Vision*, pages 206–226, 2020.

[298] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[299] 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 Communications*, 10:1096, 2019.

[300] S. Latif, M. Usman, R. Rana, and J. Qadir. Phonocardiographic sensing using deep learning for abnormal heartbeat detection. *IEEE Sensors Journal*, 18(22):9393–9400, 2018.

[301] J. Laurikkala, M. Juhola, E. Kentala, N. Lavrac, S. Miksch, and B. Kavsek. Informal identification of outliers in medical data. In *5th International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, volume 1, pages 20–24, 2000.

[302] A. Lavin and S. Ahmad. Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark. In *International Conference on Machine Learning and Applications*, pages 38–44. IEEE, 2015.

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

[304] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. J. Huang. A tutorial on energy-based learning. In G. Bakir, T. Hofman, B. Schölkopf, A. Smola, and B. Taskar, editors, *Predicting Structured Data*. MIT Press, 2006.

[305] Y. LeCun, C. Cortes, and C. J. C. Burges. MNIST handwritten digit database. *AT&T Labs*, 2, 2010.

[306] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient BackProp. In G. Montavon, G. B. Orr, and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade - Second Edition*, volume 7700 of *Lecture Notes in Computer Science*, pages 9–48. Springer, Berlin, Heidelberg, 2012.

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

[308] G. Lee and C. Scott. Nested support vector machines. *IEEE Transactions on Signal Processing*, 58(3):1648–1660, 2009.

[309] G. Lee and C. D. Scott. The one class support vector machine solution path. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 521–524, 2007.

[310] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, pages 801–808, 2007.

[311] H. Lee, P. Pham, Y. Largman, and A. Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems*, pages 1096–1104, 2009.

[312] J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer New York, 2007.

[313] J. D. Lee and K. A. See. Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1):50–80, 2004.

[314] J. Y. Lee and F. Dernoncourt. Sequential short-text classification with recurrent and convolutional neural networks. In *North American Chapter of the Association for Computational Linguistics*, pages 515–520, 2016.

[315] K. Lee, H. Lee, K. Lee, and J. Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018.

[316] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018.

[317] W. Lee and D. Xiang. Information-theoretic measures for anomaly detection. In *IEEE Symposium on Security and Privacy*, pages 130–143. IEEE, 2001.

[318] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific Reports*, 7(1):1–14, 2017.

[319] T. Lengauer, O. Sander, S. Sierra, A. Thielen, and R. Kaiser. Bioinformatics prediction of HIV coreceptor usage. *Nature Biotechnology*, 25(12):1407, 2007.

[320] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.

[321] 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 *International Conference on Artificial Neural Networks*, pages 703–716, 2019.

[322] Z. Li, N. Li, K. Jiang, Z. Ma, X. Wei, X. Hong, and Y. Gong. Superpixel masking and inpainting for self-supervised anomaly detection. In *British Machine Vision Conference*, 2020.

[323] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

[324] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16–24, 2013.

[325] 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 *Advances in Neural Information Processing Systems*, pages 4255–4265, 2019.

[326] H. Lim, J. Park, and Y. Han. Rare sound event detection using 1d convolutional recurrent neural networks. In *Workshop on Detection and Classification of Acoustic Scenes and Events*, pages 80–84, 2017.

[327] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *International Conference on Computer Vision*, pages 2980–2988, 2017.

[328] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. In *International Conference on Learning Representations*, 2017.

[329] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, Jan. 1980.

[330] R. Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

[331] Z. C. Lipton. The doctor just won't accept that! In *NeurIPS 2017 Interpretable ML Symposium*, 2017.

[332] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017.

[333] B. Liu, W. S. Lee, P. S. Yu, and X. Li. Partially supervised classification of text documents. In *International Conference on Machine Learning*, volume 2, pages 387–394, 2002.

[334] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *IEEE International Conference on Data Mining*, pages 413–422, 2008.

[335] S. Liu, R. Garrepalli, T. Dietterich, A. Fern, and D. Hendrycks. Open category detection with PAC guarantees. In *International Conference on Machine Learning*, volume 80, pages 3169–3178, 2018.

[336] W. Liu, G. Hua, and J. R. Smith. Unsupervised one-class learning for automatic outlier removal. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3826–3833, 2014.

[337] W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. J. Radke, and O. Camps. Towards visually explaining variational autoencoders. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8642–8651, 2020.

[338] Y. Liu and Y. F. Zheng. Minimum enclosing and maximum excluding machine for pattern description and discrimination. In *International Conference on Pattern Recognition*, pages 129–132, 2006.

[339] P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, M. Kloft, and K.-R. Müller. Explainable deep one-class classification. In *International Conference on Learning Representations*, 2021.

[340] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, volume 97, pages 4114–4124, 2019.

[341] F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, and M. Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 7753–7764, 2020.

[342] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[343] F. Lopez, M. Saez, Y. Shao, E. C. Balta, J. Moyne, Z. M. Mao, K. Barton, and D. Tilbury. Categorization of anomalies in smart manufacturing systems to support the selection of detection mechanisms. *Robotics and Automation Letters*, 2(4):1885–1892, 2017.

[344] W. Lu, Y. Cheng, C. Xiao, S. Chang, S. Huang, B. Liang, and T. Huang. Unsupervised sequential outlier detection with deep architectures. *IEEE Transactions on Image Processing*, 26(9):4321–4330, 2017.

[345] W. Luo, Y. Li, R. Urtasun, and R. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 4898–4906, 2016.

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

[347] M. Macha and L. Akoglu. Explaining anomalies in groups with characterizing subspace rules. *Data Mining and Knowledge Discovery*, 32(5):1444–1480, 2018.

[348] D. J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[349] D. J. C. MacKay and M. N. Gibbs. Density networks. In *Statistics and Neural Networks: Advances at the Interface*, pages 129–146. Oxford University Press, USA, 1998.

[350] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[351] A. Mahapatra, N. Srivastava, and J. Srivastava. Contextual anomaly detection in text data. *Algorithms*, 5(4):469–489, 2012.

[352] A. Makhzani and B. Frey. k-sparse autoencoders. In *International Conference on Learning Representations*, 2014.

[353] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations 2016 - Workshop Track*, 2016.

[354] 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 *International Conference on Computing, Networking and Communications*, pages 893–898, 2018.

[355] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

[356] L. M. Manevitz and M. Yousef. One-class SVMs for document classification. *Journal of Machine Learning Research*, 2(Dec):139–154, 2001.

[357] L. M. Manevitz and M. Yousef. One-class document classification via neural networks. *Neurocomputing*, 70(7):1466–1481, Mar. 2007.

[358] 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 *International Conference on Acoustics, Speech, and Signal Processing*, pages 1996–2000, 2015.

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

[360] M. Markou and S. Singh. Novelty detection: a review—part 2: neural network based approaches. *Signal Processing*, 83(12):2499–2521, Dec. 2003.

[361] L. Martí, N. Sanchez-Pi, J. M. Molina, and A. C. B. Garcia. Anomaly detection based on sensor data in petroleum industry applications. *Sensors*, 15(2):2774–2797, 2015.

[362] J. Marzat, H. Piet-Lahanier, F. Damongeot, and E. Walter. Model-based fault diagnosis for aerospace systems: a survey. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 226(10):1329–1360, 2012.

[363] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations*, 2016.

[364] F. D. Mattia, P. Galeone, M. D. Simoni, and E. Ghelfi. A survey on GANs for anomaly detection. *arXiv preprint arXiv:1906.11632*, 2019.

[365] Ł. Maziarka, M. Śmieja, M. Sendera, Ł. Struski, J. Tabor, and P. Spurek. Flow-based anomaly detection. *arXiv preprint arXiv:2010.03002*, 2020.

[366] A. Meinke and M. Hein. Towards neural networks that provably know when they don't know. In *International Conference on Learning Representations*, 2020.

[367] A. K. Menon and R. C. Williamson. A loss framework for calibrated anomaly detection. In *Advances in Neural Information Processing Systems*, pages 1494–1504, 2018.

[368] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. In *International Conference on Learning Representations*, 2017.

[369] B. Micenková, R. T. Ng, X.-H. Dang, and I. Assent. Explaining outliers by subspace separability. In *IEEE International Conference on Data Mining*, pages 518–527, 2013.

[370] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[371] 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 *International Conference on Cloud Computing and Security*, pages 322–334, 2018.

[372] S. Min, B. Lee, and S. Yoon. Deep learning in bioinformatics. *Briefings in Bioinformatics*, 18 (5):851–869, 2017.

[373] T. Minter. Single-class classification. In *LARS Symposia*, page 54, 1975.

[374] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[375] A.-r. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):14–22, 2011.

[376] G. Montavon, M. L. Braun, and K.-R. Müller. Kernel analysis of deep networks. *Journal of Machine Learning Research*, 12(Sep):2563–2581, 2011.

[377] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[378] M. M. Moya and D. R. Hush. Network constraints and multi-objective optimization for one-class classification. *Neural Networks*, 9(3):463–474, 1996.

[379] M. M. Moya, M. W. Koch, and L. D. Hostetler. One-class classifier networks for target recognition applications. In *Proceedings World Congress on Neural Networks*, pages 797–801, 1993.

[380] N. Mu and J. Gilmer. MNIST-C: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019.

[381] A. Muñoz and J. M. Moguerza. Estimation of high-density regions using one-class neighbor machines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):476–480, 2006.

[382] K. Muandet and B. Schölkopf. One-class support measure machines for group anomaly detection. In *Conference on Uncertainty in Artificial Intelligence*, pages 449–458, 2013.

[383] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[384] 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 Transactions on Geoscience and Remote Sensing*, 48(8):3188–3197, 2010.

[385] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[386] B. Nachman and D. Shih. Anomaly detection with density estimation. *Physical Review D*, 101:075042, Apr 2020.

[387] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019.

[388] E. Nalisnick, A. Matsukawa, Y. W. Teh, and B. Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using typicality. In *4th Workshop on Bayesian Deep Learning (NeurIPS 2019)*, 2019.

[389] P. Napoletano, F. Piccoli, and R. Schettini. Anomaly detection in nanofibrous materials by CNN-based self-similarity. *Sensors (Basel)*, 18(1):209, 2018.

[390] L. Naud and A. Lavin. Manifolds for unsupervised visual anomaly detection. *arXiv preprint arXiv:2006.11364*, 2020.

[391] S. Nedelkoski, J. Bogatinovski, A. Acker, J. Cardoso, and O. Kao. Self-attentive classification-based anomaly detection in unstructured logs. *arXiv preprint arXiv:2008.09340*, 2020.

[392] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[393] J. Ngiam, Z. Chen, P. W. Koh, and A. Ng. Learning deep energy models. In *International Conference on Machine Learning*, pages 1105–1112, 2011.

[394] D. T. Nguyen, Z. Lou, M. Klar, and T. Brox. Anomaly detection with multiple-hypotheses predictions. In *International Conference on Machine Learning*, volume 97, pages 4800–4809, 2019.

[395] M. H. Nguyen and F. Torre. Robust kernel principal component analysis. In *Advances in Neural Information Processing Systems*, pages 1185–1192, 2009.

[396] C. Ni, N. Charoenphakdee, J. Honda, and M. Sugiyama. On the calibration of multiclass classification with rejection. In *Advances in Neural Information Processing Systems*, pages 2586–2596, 2019.

[397] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *International Conference on Knowledge Discovery & Data Mining*, pages 631–636, 2003.

[398] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *International Conference on Computer Vision*, pages 1520–1528, 2015.

[399] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84, 2016.

[400] A. Odena. Semi-supervised learning with generative adversarial networks. In *ICML 2016 Workshop on Data Efficient Machine Learning*, 2016.

[401] E. Oja. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982.

[402] E. Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5(6):927–935, 1992.

[403] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, and I. J. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, volume 31, pages 3235–3246, 2018.

[404] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[405] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997.

[406] T. I. Oprea. Chemical space navigation in lead discovery. *Current Opinion in Chemical Biology*, 6(3):384–389, 2002.

[407] K. Ouardini, H. Yang, B. Unnikrishnan, M. Romain, C. Garcin, H. Zenati, J. P. Campbell, M. F. Chiang, J. Kalpathy-Cramer, V. Chandrasekhar, 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*, pages 225–234. Springer, 2019.

[408] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13991–14002, 2019.

[409] P. Oza and V. M. Patel. One-class convolutional neural network. *IEEE Signal Processing Letters*, 26(2):277–281, 2019.

[410] P. Oza and V. M. Patel. C2AE: Class conditioned auto-encoder for open-set recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2307–2316, 2019.

[411] G. Pang, C. Shen, and A. van den Hengel. Deep anomaly detection with deviation networks. In *International Conference on Knowledge Discovery & Data Mining*, pages 353–362, 2019.

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

[413] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

[414] D. Park, Y. Hoshi, and C. C. Kemp. A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. *Robotics and Automation Letters*, 3 (3):1544–1551, 2018.

[415] S. Park, G. Adosoglou, and P. M. Pardalos. Interpreting rate-distortion of variational autoencoder and using model uncertainty for anomaly detection. *Annals of Mathematics and Artificial Intelligence*, pages 1573–7470, 2021.

[416] L. Parra, G. Deco, and S. Miesbach. Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computation*, 8(2):260–269, 1996.

[417] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[418] A. Patcha and J.-M. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470, 2007.

[419] N. Pawlowski, M. C. Lee, M. Rajchl, S. McDonagh, E. Ferrante, K. Kamnitsas, S. Cooke, S. Stevenson, A. Khetani, T. Newman, F. Zeiler, R. Digby, J. P. Coles, D. Rueckert, D. K. Menon, V. F. J. Newcombe, and B. Glocker. Unsupervised lesion detection in brain CT using bayesian convolutional autoencoders. In *Medical Imaging with Deep Learning*, 2018.

[420] K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[421] D. Pelleg and A. W. Moore. Active learning for anomaly and rare-category detection. In *Advances in Neural Information Processing Systems*, pages 1073–1080, 2005.

[422] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[423] P. Perera and V. M. Patel. Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, 28(11):5450–5463, 2019.

[424] P. Perera, R. Nallapati, and B. Xiang. OCGAN: One-class novelty detection using GANs with constrained latent representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2898–2906, 2019.

[425] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *North American Chapter of the Association for Computational Linguistics*, pages 2227–2237, 2018.

[426] S. Pidhorskyi, R. Almohsen, and G. Doretto. Generative probabilistic novelty detection with adversarial autoencoders. In *Advances in Neural Information Processing Systems*, pages 6822–6833, 2018.

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

[428] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, April 1998.

[429] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.

[430] R. Pless and R. Souvenir. A survey of manifold learning for images. *IPSJ Transactions on Computer Vision and Applications*, 1:83–94, 2009.

[431] W. Polonik. Measuring mass concentrations and estimating density contour clusters-an excess mass approach. *The Annals of Statistics*, 23(3):855–881, 1995.

[432] W. Polonik. Minimum volume sets and generalized quantile processes. *Stochastic Processes and Their Applications*, 69(1):1–24, 1997.

[433] E. Principi, F. Vesperini, S. Squartini, and F. Piazza. Acoustic novelty detection with adversarial autoencoders. In *International Joint Conference on Neural Networks*, pages 3324–3330, 2017.

[434] P. Protopapas, J. Giammarco, L. Faccioli, M. Struble, R. Dave, and C. Alcock. Finding outlier light curves in catalogues of periodic variable stars. *Monthly Notices of the Royal Astronomical Society*, 369(2):677–696, 2006.

[435] G. Quellec, M. Lamard, M. Cozic, G. Coatrieux, and G. Cazuguel. Multiple-instance learning for anomaly detection in digital mammography. *IEEE Transactions on Medical Imaging*, 35 (7):1604–1614, 2016.

[436] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[437] A. Rabaoui, M. Davy, S. Rossignol, and N. Ellouze. Using one-class svms and wavelets for audio surveillance. *IEEE Transactions on Information Forensics and Security*, 3(4):763–775, 2008.

[438] J. Rabatel, S. Bringay, and P. Poncelet. Anomaly detection in monitoring sensor data for preventive maintenance. *Expert Systems with Applications*, 38(6):7003–7015, 2011.

[439] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[440] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. S. Dickstein. On the expressive power of deep neural networks. In *International Conference on Machine Learning*, volume 70, pages 2847–2854, 2017.

[441] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 427–438, 2000.

[442] 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*, 18 (8):2491, 2018.

[443] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova. Anomaly detection in dynamic networks: A survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(3):223–247, 2015.

[444] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[445] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282, 2017.

[446] G. Rätsch, S. Mika, B. Schölkopf, and K.-R. Müller. Constructing boosting algorithms from SVMs: An application to one-class classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1184–1199, Sep. 2002.

[447] S. Rayana. ODDS library, 2016. URL `http://odds.cs.stonybrook.edu`.

[448] A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, pages 14866–14876, 2019.

[449] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[450] 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 *Advances in Neural Information Processing Systems*, pages 14680–14691, 2019.

[451] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[452] J. Rennie. The 20 Newsgroups data set. `http://qwone.com/~jason/20Newsgroups/`, 2008. [Online; accessed 01-March-2021].

[453] E. Reyes and P. A. Estévez. Transformation based deep anomaly detection in astronomical images. *arXiv preprint arXiv:2005.07779*, 2020.

[454] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, volume 32, pages 1278–1286, 2014.

[455] C. Richter and N. Roy. Safe visual navigation via deep learning and novelty detection. In *Robotics: Science and Systems XIII*, 2017.

[456] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *International Conference on Machine Learning*, pages 833–840, 2011.

[457] P. Rigollet, R. Vert, et al. Optimal rates for plug-in estimators of density level sets. *Bernoulli*, 15(4):1154–1178, 2009.

[458] S. Roberts and L. Tarassenko. A probabilistic resource allocating network for novelty detection. *Neural Computation*, 6(2):270–284, Mar. 1994.

[459] Y. Roh, G. Heo, and S. E. Whang. A survey on data collection for machine learning: A big data - ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[460] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.

[461] V. Roth. Outlier detection with one-class kernel fisher discriminants. In *Advances in Neural Information Processing Systems*, pages 1169–1176, 2005.

[462] V. Roth. Kernel fisher discriminants for outlier detection. *Neural Computation*, 18(4):942–960, 2006.

[463] P. J. Rousseeuw. Multivariate estimation with high breakdown point. *Mathematical Statistics and Applications*, 8:283–297, 1985.

[464] P. J. Rousseeuw and K. V. Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

[465] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, 2005.

[466] L. Ruff, R. A. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *International Conference on Machine Learning*, volume 80, pages 4390–4399, 2018.

[467] 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 *ICML 2019 Workshop on Uncertainty & Robustness in Deep Learning*, 2019.

[468] L. Ruff, Y. Zemlyanskiy, R. A. Vandermeulen, T. Schnake, and M. Kloft. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In *Annual Meeting of the Association for Computational Linguistics*, pages 4061–4071, 2019.

[469] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020.

[470] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, 2021. doi: 10.1109/JPROC.2021.3052449.

[471] L. Ruff, R. A. Vandermeulen, B. J. Franks, K.-R. Müller, and M. Kloft. Rethinking assumptions in deep anomaly detection. In *ICML 2021 Workshop on Uncertainty & Robustness in Deep Learning*, 2021.

[472] 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*, chapter 8, pages 318–362. MIT Press, 1986.

[473] D. Rumsfeld. *Known and Unknown: A Memoir*. Penguin, 2011.

[474] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[475] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, 172:88–97, 2018.

[476] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli. Adversarially learned one-class classifier for novelty detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018.

[477] M. Sabokrou, M. Fathy, G. Zhao, and E. Adeli. Deep end-to-end one-class classifier. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–10, 2020.

[478] R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pages 448–455, 2009.

[479] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[480] W. Samek, S. Nakajima, M. Kawanabe, and K.-R. Müller. On robust parameter estimation in brain–computer interfacing. *Journal of Neural Engineering*, 14(6):061001, 2017.

[481] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, editors. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*. Springer, 2019.

[482] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021.

[483] N. Sarafijanovic-Djukic and J. Davis. Fast distance-based anomaly detection in images using an inception-like autoencoder. In *International Conference on Discovery Science*, pages 493–508. Springer, 2019.

[484] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox. On the information bottleneck theory of deep learning. In *International Conference on Learning Representations*, 2018.

[485] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, 2012.

[486] W. J. Scheirer, L. P. Jain, and T. E. Boult. Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2317–2324, 2014.

[487] R. T. Schirrmeister, Y. Zhou, T. Ball, and D. Zhang. Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. In *Advances in Neural Information Processing Systems*, 2020.

[488] 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 *International Conference on Information Processing in Medical Imaging*, pages 146–157, 2017.

[489] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54:30–44, 2019.

[490] J. Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.

[491] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[492] S. Schneider, A. Baevski, R. Collobert, and M. Auli. wav2vec: Unsupervised pre-training for speech recognition. In *Interspeech*, pages 3465–3469, 2019.

[493] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT press, 2002.

[494] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[495] B. Schölkopf, S. Mika, C. J. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.

[496] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[497] 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 and Knowledge Discovery*, 28(1):190–237, 2014.

[498] K. T. Schütt, F. Arbabzadah, S. Chmiela, K.-R. Müller, and A. Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8:13890, 2017.

[499] 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 Communications*, 10:5024, 2019.

[500] C. D. Scott and R. D. Nowak. Learning minimum volume sets. *Journal of Machine Learning Research*, 7(Apr):665–704, 2006.

[501] P. Seeböck, J. I. Orlando, T. Schlegl, S. M. Waldstein, H. Bogunović, S. Klimscha, G. Langs, and U. Schmidt-Erfurth. Exploiting epistemic uncertainty of anatomy segmentation for anomaly detection in retinal OCT. *IEEE Transactions on Medical Imaging*, 39(1):87–98, 2019.

[502] 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 *International Conference on Learning Representations*, 2020.

[503] G. Shalev, Y. Adi, and J. Keshet. Out-of-distribution detection using multiple semantic label representations. In *Advances in Neural Information Processing Systems*, pages 7375–7385, 2018.

[504] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July, October 1948.

[505] V. Sharan, P. Gopalan, and U. Wieder. Efficient anomaly detection via matrix sketching. In *Advances in Neural Information Processing Systems*, pages 8069–8080, 2018.

[506] L. Shen, Z. Li, and J. Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. In *Advances in Neural Information Processing Systems*, 2020.

[507] W. A. Shewhart. *Economic Control of Quality Of Manufactured Product*. Macmillan & Co. Ltd, London, 1931.

[508] L. Shu, H. Xu, and B. Liu. DOC: Deep open classification of text documents. In *Conference on Empirical Methods in Natural Language Processing*, pages 2911–2916, 2017.

[509] R. Shu, Y. Chen, A. Kumar, S. Ermon, and B. Poole. Weakly supervised disentanglement with guarantees. In *International Conference on Learning Representations*, 2020.

[510] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[511] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. In *IEEE International Conference on Data Mining*, pages 353–365, 2003.

[512] A. Siddiqui, A. Fern, T. G. Dietterich, R. Wright, A. Theriault, and D. W. Archer. Feedback-guided anomaly discovery via online optimization. In *International Conference on Knowledge Discovery & Data Mining*, pages 2200–2209, 2018.

[513] M. D. Siddiqui, A. Fern, T. G. Dietterich, and W. K. Wong. Sequential feature explanations for anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, 13(1):1–22, 2019. ISSN 1556472X. doi: 10.1145/3230666.

[514] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[515] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[516] K. Singh and S. Upadhyaya. Outlier detection: applications and techniques. *International Journal of Computer Science Issues*, 9(1):307, 2012.

[517] J. Sipple. Interpretable , multidimensional , multimodal anomaly detection with negative sampling for detection of device failure. In *International Conference on Machine Learning*, pages 4368–4377, 2020.

[518] K. Sjöstrand and R. Larsen. The entire regularization path for the support vector domain description. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 241–248, 2006.

[519] N. Slonim, G. S. Atwal, G. Tkačik, and W. Bialek. Information-based clustering. *Proceedings of the National Academy of Sciences*, 102(51):18297–18302, 2005.

[520] K. Smets, B. Verdonk, and E. M. Jordaan. Discovering novelty in spatio/temporal data using one-class support vector machines. In *International Joint Conference on Neural Networks*, pages 2956–2963, 2009.

[521] K. Sohn, C.-L. Li, J. Yoon, M. Jin, and T. Pfister. Learning and evaluating representations for deep one-class classification. In *International Conference on Learning Representations*, 2021.

[522] H. Song, Z. Jiang, A. Men, and B. Yang. A hybrid semi-supervised anomaly detection model for high-dimensional data. *Computational Intelligence and Neuroscience*, 2017.

[523] X. Song, M. Wu, C. Jermaine, and S. Ranka. Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19(5):631–645, 2007.

[524] G. Steinbuss and K. Böhm. Generating artificial outliers in the absence of genuine ones – a survey. *arXiv preprint arXiv:2006.03646*, 2020.

[525] I. Steinwart, D. Hush, and C. Scovel. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6(Feb):211–232, 2005.

[526] J. W. Stokes, J. C. Platt, J. Kravis, and M. Shilman. ALADIN: Active learning of anomalies to detect intrusions. Technical Report MSR-TR-2008-24, Microsoft Research, 2008.

[527] M. Stolpe, K. Bhaduri, K. Das, and K. Morik. Anomaly detection in vertically partitioned data by distributed Core Vector Machines. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 321–336, 2013.

[528] M. Sugiyama and M. Kawanabe. *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*. MIT press, 2012.

[529] M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(May):985–1005, 2007.

[530] S. Suh, D. H. Chae, H. Kang, and S. Choi. Echo-state conditional variational autoencoder for anomaly detection. In *International Joint Conference on Neural Networks*, pages 1015–1022, 2016.

[531] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[532] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[533] J. Tack, S. Mo, J. Jeong, and J. Shin. CSI: Novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems*, volume 33, pages 11839–11852, 2020.

[534] J. Tamboli and M. Shukla. A survey of outlier detection algorithms for data streams. In *Proceedings of the 3rd International Conference on Computing for Sustainable Global Development*, pages 3535–3540, 2016.

[535] L. Tarassenko, P. Hayton, N. Cerneaz, and M. Brady. Novelty detection for the identification of masses in mammograms. In *International Conference on Artificial Neural Networks*, pages 442–447, 1995.

[536] D. M. J. Tax. *One-Class Classification*. PhD thesis, Delft University of Technology, 2001.

[537] D. M. J. Tax and R. P. W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11):1191–1199, 1999.

[538] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1): 45–66, 2004.

[539] D. M. J. Tax and R. P. W. Duin. Growing a multi-class classifier with a reject option. *Pattern Recognition Letters*, 29(10):1565–1570, 2008.

[540] J. P. Theiler and D. M. Cai. Resampling approach for anomaly detection in multispectral images. In *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery IX*, volume 5093, pages 230–240. International Society for Optics and Photonics, 2003.

[541] S. Theodoridis. *Machine Learning: A Bayesian and Optimization Perspective*. Academic Press, 2nd edition, 2020.

[542] R. Tibshirani and T. Hastie. Outlier sums for differential gene expression analysis. *Biostatistics*, 8(1):2–8, 2007.

[543] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[544] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015.

[545] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *The 37th annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[546] S. A. Tomlins, D. R. Rhodes, S. Perner, S. M. Dhanasekaran, R. Mehra, X.-W. Sun, S. Varambally, X. Cao, J. Tchinda, R. Kuefer, et al. Recurrent fusion of TMPRSS2 and ETS transcription factor genes in prostate cancer. *Science*, 310(5748):644–648, 2005.

[547] A. Tong, G. Wolf, and S. Krishnaswamyt. Fixing bias in reconstruction-based anomaly detection with lipschitz discriminators. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2020.

[548] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[549] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.

[550] R. S. Tsay. Outliers, level shifts, and variance changes in time series. *Journal of Forecasting*, 7(1):1–20, 1988.

[551] R. S. Tsay, D. Peña, and A. E. Pankratz. Outliers in multivariate time series. *Biometrika*, 87 (4):789–804, Dec. 2000.

[552] M. Tschannen, O. Bachem, and M. Lucic. Recent advances in autoencoder-based representation learning. In *3rd Workshop on Bayesian Deep Learning (NeurIPS 2018)*, 2018.

[553] A. B. Tsybakov. On nonparametric estimation of density level sets. *The Annals of Statistics*, 25(3):948–969, 1997.

[554] N. Tuluptceva, B. Bakker, I. Fedulova, H. Schulz, and D. V. Dylov. Anomaly detection with deep perceptual autoencoders. *arXiv preprint arXiv:2006.13265*, 2020.

[555] 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. *International Journal of Accounting Information Aystems*, 21:18–31, 2016.

[556] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.

[557] L. van der Maaten, E. Postma, and J. van den Herik. Dimensionality reduction: A comparative review. Technical Report TiCC-TR 2009-005, Tilburg centre for Creative Computing (TiCC), Tilburg University, 2009.

[558] R. Vandermeulen and C. Scott. Consistency of robust kernel density estimators. In *Conference on Learning Theory*, pages 568–591, 2013.

[559] R. A. Vandermeulen, R. Saitenmacher, and A. Ritchie. A proposal for supervised density estimation. In *NeurIPS 2020 Pre-registration Workshop*, 2020.

[560] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[561] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis. Attention guided anomaly localization in images. In *European Conference on Computer Vision*, pages 485–503, 2020.

[562] J. Verbesselt, A. Zeileis, and M. Herold. Near real-time disturbance detection using satellite image time series. *Remote Sensing of Environment*, 123:98–108, 2012.

[563] R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*, volume 47. Cambridge University Press, 2018.

[564] R. Vert and J.-P. Vert. Consistency and convergence rates of one-class SVMs and related algorithms. *Journal of Machine Learning Research*, 7(May):817–854, 2006.

[565] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, pages 1096–1103, 2008.

[566] 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. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[567] N. X. Vinh, J. Chan, S. Romano, J. Bailey, C. Leckie, K. Ramamohanarao, and J. Pei. Discovering outlying aspects in large datasets. *Data Mining and Knowledge Discovery*, 30(6): 1520–1555, 2016.

[568] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites. *Journal für die Reine und Angewandte Mathematik*, 1908(133):97–178, 1908.

[569] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les paralléllοèdres primitifs. *Journal für die Reine und Angewandte Mathematik*, 1908(134):198–287, 1908.

[570] S. Walfish. A review of statistical outlier methods. *Pharmaceutical Technology*, 30(11):1–5, 2006.

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

[572] J. Wang and A. Cherian. GODS: Generalized one-class discriminative subspaces for anomaly detection. In *International Conference on Computer Vision*, pages 8201–8211, 2019.

[573] J. Wang, P. Neskovic, and L. N. Cooper. Pattern classification via single spheres. In *International Conference on Discovery Science*, pages 241–252. Springer, 2005.

[574] J. Wang, S. Sun, and Y. Yu. Multivariate triangular quantile maps for novelty detection. In *Advances in Neural Information Processing Systems*, pages 5061–5072, 2019.

[575] S. Wang, Y. Zeng, X. Liu, E. Zhu, J. Yin, C. Xu, and M. Kloft. Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. In *Advances in Neural Information Processing Systems*, pages 5960–5973, 2019.

[576] T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, volume 119, pages 9929–9939, 2020.

[577] L. Wellhausen, R. Ranftl, and M. Hutter. Safe robot navigation via multi-modal anomaly detection. *Robotics and Automation Letters*, 5(2):1326–1333, 2020.

[578] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning*, pages 681–688, 2011.

[579] T. Wiatowski and H. Bölcskei. A mathematical theory of deep convolutional neural networks for feature extraction. *IEEE Transactions on Information Theory*, 64(3):1845–1866, 2018.

[580] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu. Towards universal paraphrastic sentence embeddings. In *International Conference on Learning Representations*, 2016.

[581] J. Winkens, R. Bunel, A. G. Roy, R. Stanforth, V. Natarajan, J. R. Ledsam, P. MacWilliams, P. Kohli, A. Karthikesalingam, S. Kohl, et al. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.

[582] W.-K. Wong, A. W. Moore, G. F. Cooper, and M. M. Wagner. Bayesian network anomaly pattern detection for disease outbreaks. In *International Conference on Machine Learning*, pages 808–815, 2003.

[583] 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. *Journal of Machine Learning Research*, 6(Dec):1961–1998, 2005.

[584] 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 *Annual Meeting of the Association for Computational Linguistics*, pages 808–819, 2019.

[585] P. Wu, J. Liu, and F. Shen. A deep one-class neural network for anomalous event detection in complex scenes. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7): 2609–2622, 2020.

[586] R. Wu and E. J. Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *arXiv preprint arXiv:2009.13807*, 2020.

[587] 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 Transactions on Geoscience and Remote Sensing*, 57(1):62–75, 2018.

[588] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *International Conference on Computer Vision*, pages 1511–1519, 2015.

[589] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[590] Y. Xiao, H. Wang, W. Xu, and J. Zhou. L1 norm based KPCA for novelty detection. *Pattern Recognition*, 46(1):389–396, 2013.

[591] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, volume 48, pages 478–487, 2016.

[592] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

[593] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang. Machine learning and deep learning methods for cybersecurity. *IEEE Access*, 6:35365–35381, 2018.

[594] L. Xiong, B. Póczos, and J. G. Schneider. Group anomaly detection using flexible genre models. In *Advances in Neural Information Processing Systems*, pages 1071–1079, 2011.

[595] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In *World Wide Web Conference*, pages 187–196, 2018.

[596] X. Xu, H. Liu, and M. Yao. Recent progress of anomaly detection. *Complexity*, 2019:1–11, 2019.

[597] W. Yan and L. Yu. On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach. In *Annual Conference of the Prognostics and Health Management Society*, volume 6, 2015.

[598] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *International Conference on Machine Learning*, volume 70, pages 3861–3870, 2017.

[599] T. Y. Yap. Text anomaly detection with arae-anogan. Honors Projects 22, Illinois Wesleyan University, 2020.

[600] S. You, K. C. Tezcan, X. Chen, and E. Konukoglu. Unsupervised lesion detection via image restoration with a normative prior. In *International Conference on Medical Imaging with Deep Learning*, pages 540–556, 2019.

[601] R. Yu, X. He, and Y. Liu. GLAD: Group anomaly detection in social media analysis. *ACM Transactions on Knowledge Discovery from Data*, 10(2):1–22, 2015.

[602] S. Zagoruyko and N. Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

[603] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833, 2014.

[604] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. Chandrasekhar. Adversarially learned anomaly detection. In *IEEE International Conference on Data Mining*, pages 727–736, 2018.

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

[606] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang. Deep structured energy based models for anomaly detection. In *International Conference on Machine Learning*, volume 48, pages 1100–1109, 2016.

[607] B. Zhang and W. Zuo. Learning from positive and unlabeled examples: A survey. In *Proceedings of the IEEE International Symposium on Information Processing*, pages 650–654, 2008.

[608] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

[609] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 1409–1416, 2019.

[610] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, volume 97, pages 7472–7482, 2019.

[611] H. Zhang, A. Li, J. Guo, and Y. Guo. Hybrid models for open set recognition. *arXiv preprint arXiv:2003.12506*, 2020.

[612] J. Zhang. Advancements of outlier detection: A survey. *ICST Transactions on Scalable Information Systems*, 13(1):1–26, 2013.

[613] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666, 2016.

[614] M. Zhao and V. Saligrama. Anomaly detection with score functions based on nearest neighbor graphs. In *Advances in Neural Information Processing Systems*, pages 2250–2258, 2009.

[615] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115:213–237, 2019.

[616] S. Zhao, J. Song, and S. Ermon. InfoVAE: Balancing learning and inference in variational autoencoders. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 5885–5892, 2019.

[617] 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 Networks*, 102:78–86, 2018.

[618] C. Zhou and R. C. Paffenroth. Anomaly detection with robust deep autoencoders. In *International Conference on Knowledge Discovery & Data Mining*, pages 665–674, 2017.

[619] K. Zhou, Y. Xiao, J. Yang, J. Cheng, W. Liu, W. Luo, Z. Gu, J. Liu, and S. Gao. Encoding structure-texture relation with p-net for anomaly detection in retinal images. In *European Conference on Computer Vision*, 2020.

[620] Z.-H. Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5 (1):44–53, 2018.

[621] A. Zimek, E. Schubert, and H.-P. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012.

[622] D. Zimmerer, J. Petersen, G. Köhler, P. Jäger, P. Full, T. Roß, T. Adler, A. Reinke, L. Maier-Hein, and K. Maier-Hein. Medical out-of-distribution analysis challenge, Mar. 2020. URL `https://doi.org/10.5281/zenodo.3784230`.

[623] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.