Deep Support Vector Data Description for Unsupervised and Semi-Supervised Anomaly Detection

Lukas Ruff [1] Robert A. Vandermeulen [2] Nico Görnitz [1] Alexander Binder [3] Emmanuel Müller [4] Marius Kloft [2]

Abstract

Deep approaches to anomaly detection have recently shown promising results over shallow detectors on large and high-dimensional data. Most of these approaches view this task as an unsupervised learning problem. In practice however, one may have—in addition to a large set of unlabeled samples—access to a small pool of labeled samples, e.g. samples verified by some domain expert. Semi-supervised approaches to anomaly detection make use of such labeled data to improve detection performance, but so far only few, domain-specific deep methods have been proposed for semi-supervised anomaly detection. In this work, we present a generalization of the recently introduced *Deep Support Vector Data Description* method from the unsupervised to the more general semi-supervised anomaly detection setting. We demonstrate experimentally that our method consistently outperforms both deep unsupervised and deep supervised baselines on MNIST, Fashion-MNIST, and CIFAR-10, even when provided with only small amounts of labeled training data.

1. Introduction

Anomaly detection (AD) (Chandola et al., 2009; Pimentel et al., 2014) is the task of identifying unusual samples in data. This task lacks a supervised learning objective and AD methods typically formulate an unsupervised problem to find a "compact" description of the "normal" class, e.g. finding a set of small measure that contains most of the data as in one-class classification (Moya et al., 1993). Samples that deviate from this description are deemed anomalous. The limitations of shallow AD methods such as the One-Class SVM (Schölkopf et al., 2001), Support Vector Data Description (SVDD) (Tax & Duin, 2004), Isolation Forest (Liu et al., 2008), or Kernel Density Estimation (Parzen, 1962; Kim & Scott, 2012; Vandermeulen & Scott, 2013) in their scalability to large datasets and their need for manual feature engineering motivated research on novel deep approaches to AD that recently have shown promising results (Sakurada & Yairi, 2014; Erfani et al., 2016; Zhai et al., 2016; Chen et al., 2017; Ruff et al., 2018; Deecke et al., 2018; Golan & El-Yaniv, 2018; Hendrycks et al., 2019).

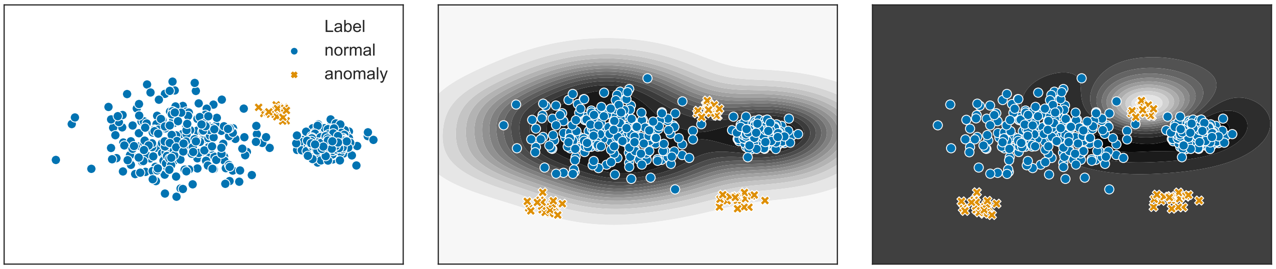
In many real-world applications verified (i.e., labeled) normal or anomalous examples are often available, in addition to a large set of unlabeled data. Such samples could be hand labeled by a domain expert, for example. An unsupervised approach would ignore this valuable information. A *fully* supervised approach to AD, on the other hand, learns to separate the anomalies from the normal data. This works well when the anomalies at test time are drawn from the same distribution as in training. In practice however, this is rarely the case: for instance in computer security attacks are generated adversarially. Figure 1 illustrates this situation on a toy example.

Semi-supervised approaches (Wang et al., 2005; Liu & Zheng, 2006; Blanchard et al., 2010; Muñoz-Marí et al., 2010; Görnitz et al., 2013) aim to bridge the gap between supervised and unsupervised AD. These approaches do not assume some common pattern among the "anomaly class" and thus do not impose the typical cluster assumption semi-supervised classifiers build upon (Zhu, 2008; Chapelle et al., 2009). Instead, semi-supervised approaches to AD aim to find a "compact description" while still correctly classifying the labeled data. Through this, semi-supervised AD methods do not overfit to the labeled anomalies and generalize to novel anomalies (Görnitz et al., 2013).

Existing work on *deep* semi-supervised learning has mainly focused on the classification task (Kingma et al., 2014; Rasmus et al., 2015; Odena, 2016; Dai et al., 2017; Oliver et al., 2018). So far, only a few deep semi-supervised approaches to AD have been proposed, most of which are domain or data-type specific (Ergen et al., 2017; Kiran et al., 2018; Min et al., 2018).

[1] Machine Learning Group, Department of Electrical Engineering & Computer Science, TU Berlin, Berlin, Germany [2] Department of Computer Science, TU Kaiserslautern, Kaiserslautern, Germany [3] ISTD pillar, Singapore University of Technology and Design, Singapore [4] Bonn-Aachen International Center for Information Technology, Bonn, Germany. Correspondence to: Lukas Ruff <contact@lukasruff.com>.

Presented at the ICML 2019 Workshop on Uncertainty and Robustness in Deep Learning. Copyright 2019 by the author(s).



(a) Training data (b) Unsupervised One-Class Model (c) Supervised Model

Figure 1. The need for semi-supervised AD methods: We consider a setting with only one known anomaly class (orange) at training time (illustrated in (a)) and two new unknown anomaly classes appearing at testing time (bottom left and bottom right of Figure (b) and (c)). The purely unsupervised method (shown in (b)) ignores the known anomalies, which are deemed normal. The purely supervised approach (shown in (c)) overfits to the previously seen anomalies but fails to generalize to the novel anomalies.

2. Deep Support Vector Data Description

Here, we introduce a generalization of *Deep Support Vector Data Description (Deep SVDD)* to the more general semi-supervised AD setting that contains the unsupervised Deep SVDD method (Ruff et al., 2018) as a special case.

2.1. Unsupervised Deep SVDD

For input space $\mathcal{X} \subseteq \mathbb{R}^d$ and output space $\mathcal{F} \subseteq \mathbb{R}^p$, let $\phi(\,\cdot\,;\mathcal{W}) : \mathcal{X} \to \mathcal{F}$ be a neural network with $L \in \mathbb{N}$ hidden layers and weights $\mathcal{W} = \{\boldsymbol{W}^1, \dots, \boldsymbol{W}^L\}$. The objective of Deep SVDD is to learn a neural network transformation ϕ that minimizes the volume of a data-enclosing hypersphere with radius $R > 0$ and fixed center $\boldsymbol{c} \in \mathcal{F}$ in output space $\mathcal{F}$. Given $n \in \mathbb{N}$ (unlabeled) training samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X}$, the *Soft-Boundary Deep SVDD* objective is defined by

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

Points mapped outside the sphere ($\|\phi(\boldsymbol{x}_i;\mathcal{W}) - \boldsymbol{c}\|^2 > R^2$) get penalized and the network weights $\mathcal{W}$ are optimized such that most of the data falls within the hypersphere centered at $\boldsymbol{c}$. Minimizing the volume of the sphere via R^2 enforces this learning process. In consequence, normal points get closely mapped to the hypersphere center, whereas anomalies are mapped further away or outside the sphere. Hyperparameter $\nu \in (0, 1]$ controls this trade-off between volume and boundary violations (Ruff et al., 2018).

If the unlabeled training data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ is not polluted, i.e. if most of the training examples are normal, the simplified *One-Class Deep SVDD* objective, which penalizes the mean squared distance of *all* the mapped data points (not just the outliers), is preferable:

$$\min_{\mathcal{W}} \quad \frac{1}{n} \sum_{i=1}^{n} \|\phi(\boldsymbol{x}_i;\mathcal{W}) - \boldsymbol{c}\|^2. \qquad (2)$$

2.2. Semi-Supervised Deep SVDD

Now we assume we also have access to $m \in \mathbb{N}$ labeled samples $(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \dots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y}$ in addition to the $n \in \mathbb{N}$ unlabeled samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X}$ with $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{-1, +1\}$. We denote $\tilde{y} = +1$ for known normal examples and $\tilde{y} = -1$ for known anomalies.

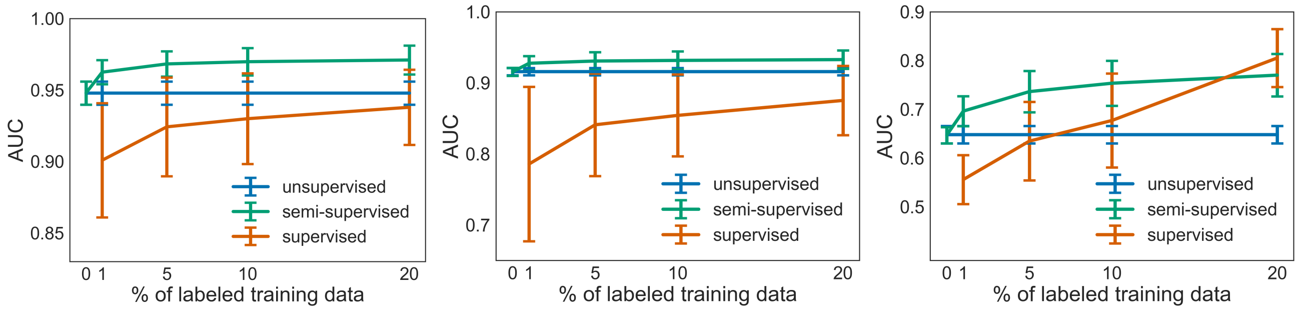
We establish a *Semi-Supervised Deep SVDD (SS-DSVDD)* generalization by extending the objectives (1) and (2) with terms that enables learning from labeled data. We formulate the *Soft-Boundary SS-DSVDD* problem as

$$\begin{aligned} \min_{R,\mathcal{W}} \quad & R^2 + \frac{1}{\nu(n+m)} \sum_{i=1}^{n} l\left(R^2 - \|\phi(\boldsymbol{x}_i;\mathcal{W}) - \boldsymbol{c}\|^2\right) \\ & + \frac{\eta}{\nu(n+m)} \sum_{j=1}^{m} l\left(\tilde{y}_j\left(R^2 - \|\phi(\tilde{\boldsymbol{x}}_j;\mathcal{W}) - \boldsymbol{c}\|^2\right)\right), \end{aligned} \qquad (3)$$

where $l(z) = \max\{0, -z\}$ is the hinge loss. That is, we require normal examples ($\tilde{y} = +1$) to lie inside the hypersphere and labeled anomalies ($\tilde{y} = -1$) to lie outside. We achieve this by penalizing accordingly: if a labeled anomaly lies *inside* the sphere, the penalty is given by $R^2 - \|\phi(\tilde{\boldsymbol{x}}_j;\mathcal{W}) - \boldsymbol{c}\|^2$ and $\|\phi(\tilde{\boldsymbol{x}}_j;\mathcal{W}) - \boldsymbol{c}\|^2 - R^2$ elsewise. If a labeled data point is already mapped onto the correct side, there is no penalty. To generalize (2), we propose the following *One-Class SS-DSVDD* objective:

$$\begin{aligned} \min_{\mathcal{W}} \quad & \frac{1}{n+m} \sum_{i=1}^{n} \|\phi(\boldsymbol{x}_i;\mathcal{W}) - \boldsymbol{c}\|^2 \\ & + \frac{\eta}{n+m} \sum_{j=1}^{m} \left(\|\phi(\tilde{\boldsymbol{x}}_j;\mathcal{W}) - \boldsymbol{c}\|^2\right)^{\tilde{y}_j}. \end{aligned} \qquad (4)$$

Here, we impose a quadratic loss on the distances of the mapped points to the fixed center $\boldsymbol{c}$, for both the unlabeled samples and the labeled normal points. For the labeled anomalies, we penalize the inverse such that anomalies must



(a) MNIST

(b) Fashion-MNIST

(c) CIFAR-10

Figure 2. Experimental results when gradually increasing the ratio of labeled training examples. We report the average AUC with standard deviation computed over the 10 AD setups with 10 seeds per setup (i.e. overall 100 runs) at various ratios for the approaches. Our Semi-Supervised Deep SVDD shows significant improvements already with small amounts of labeled data.

be mapped further away from the center.[1] In both semi-supervised objectives (3) and (4), the hyperparameter $\eta > 0$ controls the balance between the labeled and unlabeled term. For the case that only unlabeled data is available ($m = 0$), we recover (1) and (2) from (3) and (4) respectively.

The Deep SVDD anomaly score is then given by the distance to the center of the hypersphere: $s(\boldsymbol{x}) = \|\phi(\boldsymbol{x}; \mathcal{W}^*) - \boldsymbol{c}\|$. We optimize all four non-convex objectives (1)–(4) via SGD using backpropagation, where we add weight decay regularization for improved generalization. We provide further details on the optimization in Appendix A.

3. Experiments

We evaluate SS-DSVDD on MNIST, Fashion-MNIST, and CIFAR-10. Our focus in the evaluation lies on the semi-supervised setting and the detection performance in specific experimental scenarios. We compare our semi-supervised method to the corresponding natural ends on the learning spectrum: the unsupervised Deep SVDD and a fully supervised deep classifier. To control for architectural effects, we always employ the same underlying deep network $\phi(\cdot\,; \mathcal{W}) : \mathcal{X} \to \mathcal{F}$ for all three methods. Appendix B and C contain additional details regarding architecture and competitors. For a comparison of various deep anomaly detectors we refer to other recent works (Ruff et al., 2018; Golan & El-Yaniv, 2018; Hendrycks et al., 2019).

3.1. Semi-Supervised Anomaly Detection Setup

All three datasets have ten classes from which we derive ten AD setups on each dataset. In every setup, we consider one of the ten classes to be the normal class whereas samples from the remaining nine classes represent anomalies. The original training data of the respective normal class forms the unlabeled part of our training set. The training data of the respective nine anomaly classes forms the data pool from which we draw anomalies for training. We use the AUC metric to quantitatively evaluate the detection performance of the different approaches on the original respective test sets using ground truth labels, i.e. $\tilde{y} = +1$ for the normal class and $\tilde{y} = -1$ for the respective nine anomaly classes.

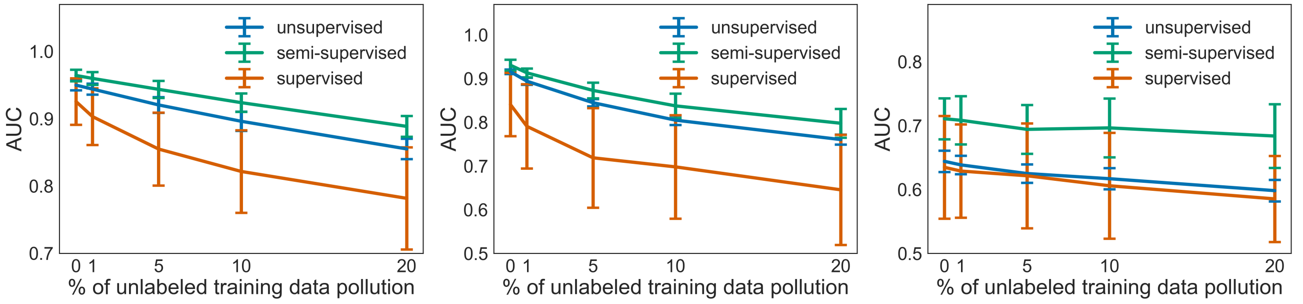
3.2. Experimental Scenarios

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

(i) Adding labeled anomalies We gradually increase the ratio of labeled training data $m/(n+m)$ by adding additional known anomalies $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_m$ with $\tilde{y}_j = -1$ to the training set. In each of the ten AD setups, we take the training data of the respective normal class for the unlabeled part of the training set and then add labeled anomalies from one a priori randomly drawn anomaly class (out of the nine remaining ones) at training time. At testing time, we always consider all nine remaining classes as anomalies, i.e. there are eight novel classes at testing. This setup was chosen to highlight the performance on out-of-distribution, novel anomalies. Note that the unlabeled part of the training set is unpolluted. We repeat this training set generation process for multiple seeds.

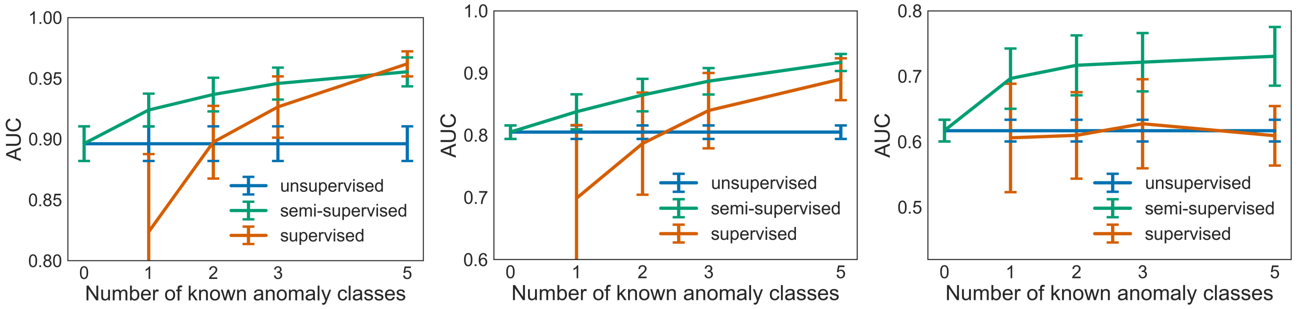
(ii) Polluted training data In this setup, we gradually pollute the unlabeled part of the training set with (unknown) anomalies drawn from all nine respective anomaly classes in each AD setup. We again repeat experiments for multiple seeds in each of the ten AD setups. In these experiments, we fix the ratio of labeled training samples at 5% which are again sampled only from one previously drawn anomaly class in every seed. We hypothesize that the semi-supervised approach alleviates the negative impact pollution has on detection performance, since labeled anomalies should help to "filter out" similar unknown anomalies.

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



(a) MNIST (b) Fashion-MNIST (c) CIFAR-10

Figure 3. Experimental results when gradually polluting the unlabeled part of the training set with anomalies. We report the average AUC with standard deviation computed over the 10 AD setups with 10 seeds per setup (i.e. overall 100 runs) at various ratios for the approaches. Our Semi-Supervised Deep SVDD methods proves to be the most robust towards training set pollution.



(a) MNIST (b) Fashion-MNIST (c) CIFAR-10

Figure 4. Experimental results when gradually increasing the number of known anomaly classes. We report the average AUC with standard deviation computed over the 10 AD setups with 10 seeds per setup (i.e. overall 100 runs) at various numbers of known anomaly classes for the approaches. The more anomaly classes are given at training time, the better Semi-Supervised Deep SVDD performs.

(iii) Increasing the number of known anomaly classes In the last scenario, we compare performance for an increasing number of known anomaly classes. In (i) and (ii), we always sample labeled anomalies for the training set from only one of the nine anomaly classes per seed in each AD setup. Here, we now gradually increase the number of anomaly classes the labeled anomalies are drawn from for the training set. Since we have a limited number of anomaly classes (nine) in our setups, we expect the supervised classifier to catch up with the semi-supervised approach at some point. We fix the overall ratio of labeled training examples again at 5% and consider a pollution ratio of 10% for the unlabeled part of the training set in this scenario.

3.3. Results and Discussion

The results of the experimental scenarios (i)–(iii) are shown in Figures 2–4. We see significant improvements in detection performance for SS-DSVDD over the unsupervised baseline already with only little labeled data in Figure 2. In comparison to the supervised classifier, which is vulnerable to novel anomalies at testing, our semi-supervised method generalizes well to novel anomalies. Figure 3 confirms that performance drops for all methods as pollution increases, where SS-DSVDD is the most robust. Finally, Figure 4 demonstrates that the more diverse the known, labeled anomalies in the training set are, the better the detection performance becomes. We see that the performance of the supervised approach is very sensitive to the number of anomaly classes, but since the number of anomaly classes is limited in our setups, the classifier catches up at some point. However, on CIFAR-10 5% labeled training data seems to be insufficient to represent the variation in the anomaly classes, which explains the bad supervised performance even at a high number of known anomaly classes. We give detailed results of all the variants in Appendix D.

4. Conclusion

We have generalized Deep SVDD to the more general semi-supervised setting in this work. The resulting Semi-Supervised Deep SVDD is an end-to-end deep method for semi-supervised anomaly detection on high-dimensional data. We demonstrated experimentally, that SS-DSVDD significantly improves detection performance already with only small amounts of labeled data. Our results suggest that semi-supervised approaches to AD should be preferred in applications where some labeled information is available.

Acknowledgements

We thank Klaus-Robert Müller for helpful comments and discussions. LR acknowledges support from the German Federal Ministry of Education and Research (BMBF) in the project ALICE III (FKZ: 01IS18049B). MK and RV acknowledge support by the German Research Foundation (DFG) award KL 2698/2-1 and by the German Federal Ministry of Education and Research (BMBF) awards 031L0023A, 01IS18051A, and 031B0770E. NG was supported by the Berlin Center for Machine Learning (BMBF grant 01IS18037I). AB is grateful for support by the Singapore Ministry of Education grant MOE2016-T2-2-154.

References

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

Chandola, V., Banerjee, A., and Kumar, V. Anomaly Detection: A Survey. *ACM Computing Surveys*, 41(3):1–58, 2009.

Chapelle, O., Schölkopf, B., and Zien, A. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3): 542–542, 2009.

Chen, J., Sathe, S., Aggarwal, C., and Turaga, D. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 90–98, 2017.

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

Deecke, L., Vandermeulen, R. A., Ruff, L., Mandt, S., and Kloft, M. Image anomaly detection with generative adversarial networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 3–17, 2018.

Erfani, S. M., Rajasegarar, S., Karunasekera, S., and Leckie, C. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, 58:121–134, 2016.

Ergen, T., Mirza, A. H., and Kozat, S. S. Unsupervised and semi-supervised anomaly detection with LSTM neural networks. *arXiv:1710.09207*, 2017.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

Golan, I. and El-Yaniv, R. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, pp. 9758–9769, 2018.

Görnitz, N., Kloft, M., Rieck, K., and Brefeld, U. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 46:235–262, 2013.

Hendrycks, D., Mazeika, M., and Dietterich, T. G. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

Ioffe, S. and Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*, volume 37, pp. 448–456, 2015.

Kim, J. and Scott, C. D. Robust kernel density estimation. *Journal of Machine Learning Research*, 13(Sep):2529–2565, 2012.

Kingma, D. and Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*, 2014.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pp. 3581–3589, 2014.

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

Liu, F. T., Ting, K. M., and Zhou, Z.-H. Isolation Forest. In *International Conference on Data Mining*, pp. 413–422, 2008.

Liu, Y. and Zheng, Y. F. Minimum enclosing and maximum excluding machine for pattern description and discrimination. In *International Conference on Pattern Recognition*, pp. 129–132, 2006.

Min, E., Long, J., Liu, Q., Cui, J., Cai, Z., and Ma, J. SU-IDS: A semi-supervised and unsupervised framework for network intrusion detection. In *International Conference on Cloud Computing and Security*, pp. 322–334, 2018.

Moya, M. M., Koch, M. W., and Hostetler, L. D. One-class classifier networks for target recognition applications. In *Proceedings World Congress on Neural Networks*, pp. 797–801, 1993.

Muñoz-Marí, J., Bovolo, F., Gómez-Chova, L., Bruzzone, L., and Camp-Valls, G. 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.

Odena, A. Semi-supervised learning with generative adversarial networks. *arXiv:1606.01583*, 2016.

Oliver, A., Odena, A., Raffel, C. A., Cubuk, E. D., and Goodfellow, I. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pp. 3235–3246, 2018.

Parzen, E. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3): 1065–1076, 1962.

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

Rasmus, A., Berglund, M., Honkala, M., Valpola, H., and Raiko, T. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pp. 3546–3554, 2015.

Ruff, L., Vandermeulen, R. A., Görnitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. Deep one-class classification. In *International Conference on Machine Learning*, volume 80, pp. 4390–4399, 2018.

Sakurada, M. and Yairi, T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the 2nd MLSDA Workshop*, pp. 4, 2014.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. Estimating the Support of a High-Dimensional Distribution. *Neural Computation*, 13(7): 1443–1471, 2001.

Tax, D. M. J. and Duin, R. P. W. Support Vector Data Description. *Machine Learning*, 54(1):45–66, 2004.

Vandermeulen, R. and Scott, C. Consistency of robust kernel density estimators. In *Conference on Learning Theory*, pp. 568–591, 2013.

Wang, J., Neskovic, P., and Cooper, L. N. Pattern classification via single spheres. In *International Conference on Discovery Science*, pp. 241–252. Springer, 2005.

Zhai, S., Cheng, Y., Lu, W., and Zhang, Z. Deep structured energy based models for anomaly detection. In *International Conference on Machine Learning*, volume 48, pp. 1100–1109, 2016.

Zhu, X. Semi-supervised learning literature survey. *Computer Sciences TR 1530, University of Wisconsin Madison*, 2008.

A. Semi-Supervised Deep SVDD Optimization

The two SS-DSVDD objectives (3) and (4) are generally non-convex in the network weights $\mathcal{W}$ which usually is the case in deep learning. We rely on (mini-batch) SGD to optimize the network weights using backpropagation. For Soft-Boundary SS-DSVDD, it would be inefficient to also update radius R via SGD using some shared learning rate, since the network parameters $\mathcal{W}$ and R generally are on different scales. Instead, analogously to Ruff et al. (2018), we suggest an alternating minimization approach. First, we update the network weights $\mathcal{W}$ using SGD keeping radius R fixed; then, given the most recent network representations of the data, we directly solve for radius R (e.g. via line search). To save some computational load, we suggest to update R on the mini-batches. With this approximation, we empirically found similar results but avoid forward passes on the full training data. For improved generalization, we add ℓ_2 weight decay regularization with hyperparameter $\lambda > 0$ to the objectives. Algorithm 1 summarizes the SS-DSVDD optimization routine. For One-Class SS-DSVDD, hyperparameter ν and radius R are dropped from the algorithm and only the network weights $\mathcal{W}$ are updated via SGD.

Algorithm 1 Optimization of SS-DSVDD

```
Input:
    Unlabeled data: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$
    Labeled data: $(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \dots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m)$
    Hyperparameters: $\nu, \eta, \lambda$
    SGD learning rate: $\varepsilon$
Output:
    Trained model: $(R^*, \mathcal{W}^*)$

Initialize:
    Neural network weights: $\mathcal{W}$
    Hypersphere parameters: $R, \boldsymbol{c}$
for each epoch do
  for each mini-batch do
    Draw mini-batch $\mathcal{B}$
    $\mathcal{W} \leftarrow \mathcal{W} - \varepsilon \cdot \nabla_{\mathcal{W}} J(R, \mathcal{W}; \mathcal{B})$
    Solve for $R$ on mini-batch $\mathcal{B}$
  end for
end for
```

Using SGD allows SS-DSVDD to scale with large datasets as the computational complexity scales linearly in the number of training batches and computations in each batch can be parallelized (e.g. by training on GPUs). SS-DSVDD also has low memory complexity as a trained model is fully characterized by the final parameters $(R^*, \mathcal{W}^*)$ and no data must be saved or referenced for prediction. Instead, the prediction only requires a forward pass on the network which usually is just a concatenation of simple functions.

Initialization of network weights $\mathcal{W}$ We empirically found the best results by establishing an autoencoder pretraining routine for initialization. That is, we first train an autoencoder that has an encoder with the same architecture as network ϕ on the reconstruction loss using only the unlabeled training data. After training, we then initialize $\mathcal{W}$ with the converged parameters of the encoder.

Initialization of center $\boldsymbol{c}$ and radius R After initializing the network weights $\mathcal{W}$, we fix the hypersphere center $\boldsymbol{c}$ as the mean of the network representations that we obtain from an initial forward pass on the data (excluding labeled anomalies). As also observed in Ruff et al. (2018), we found SGD convergence to be smoother and faster by fixing center $\boldsymbol{c}$ in the neighborhood of the initial data representations. If many labeled normal examples are available, using only those examples for a mean initialization would be another strategy to minimize distortions from polluted unlabeled training data. Radius R can be initialized with $R = 0$, for example, which emphasizes unlabeled and labeled normal samples in the beginning of the learning procedure. Adding center $\boldsymbol{c}$ to the optimization variables would allow a trivial "hypersphere collapse" solution for Deep SVDD.

Preventing a hypersphere collapse A "hypersphere collapse" describes the trivial solution that the neural network ϕ converges to the constant function $\phi \equiv \boldsymbol{c}$, i.e. the hypersphere collapses to a single point. In Ruff et al. (2018), we demonstrate theoretical network properties that prevent such a collapse which we adopt for SS-DSVDD. Most importantly, network ϕ must have no bias terms and no bounded activation functions. We refer to Ruff et al. (2018) for further details.

B. Network Architectures

We employ LeNet-type convolutional neural networks (CNNs) for all datasets, 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 dense 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 dense 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 dense layer of 128 units. For Deep SVDD, we remove all bias terms from the network to prevent a hypersphere collapse.

C. Details on Competing Methods

Unsupervised Deep SVDD Baseline We consider both variants, Soft-Boundary Deep SVDD and One-Class Deep SVDD as unsupervised baselines and always report the better performance as the unsupervised result. 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 $\lambda = 10^{-6}$.

Semi-Supervised Deep SVDD We also consider both of our SS-DSVDD objectives and again report the better performance as the semi-supervised result. For Soft-Boundary SS-DSVDD, we also run experiments for $\nu \in \{0.01, 0.1\}$. Again, we set $\lambda = 10^{-6}$. We equally weight unlabeled and labeled examples by setting $\eta = 1$.

Supervised Deep Binary Classifier To interpret AD 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.

SGD Optimization Details We use the Adam optimizer with recommended default hyperparameters (Kingma & Ba, 2014) and apply Batch Normalization (Ioffe & Szegedy, 2015) in SGD optimization. For all three approaches and on all datasets, we employ a two-phase ("searching" and "fine-tuning") learning rate schedule. In the searching phase we first train with a learning rate $\varepsilon = 10^{-4}$ for 50 epochs. In the fine-tuning phase we train with $\varepsilon = 10^{-5}$ for another 100 epochs. We always use a batch size of 200. For the supervised classifier, we initialize the network with uniform Glorot weights (Glorot & Bengio, 2010). For unsupervised and semi-supervised Deep SVDD, we establish an unsupervised pre-training routine via convolutional autoencoder (CAE) as explained in Appendix A. We set the network ϕ to be the encoder of the CAE that we train beforehand, and symmetrically construct the decoder where we replace max-pooling with simple upsampling. After training the CAE on the MSE reconstruction loss, we use the resulting encoder weights for initialization.

D. Detailed Tables of Experimental Results

Below we give detailed results of the experiments for all five methods considered: the unsupervised Soft-Boundary Deep SVDD and One-Class Deep SVDD, our semi-supervised Soft-Boundary SS-DSVDD and One-Class SS-DSVDD, as well as the supervised binary classifier. Table 1 lists the results from the experimental scenario (i) where we gradually increase the proportion of labeled training data. Table 2 lists the results from the experimental scenario (ii) with polluted unlabeled training data. Table 3 lists the results from the experimental scenario (iii) where we gradually increase the number of anomaly classes from which we draw the labeled anomalies from.

Table 1. Detailed experimental results when gradually increasing the ratio of labeled training examples. We report the average AUC with standard deviation computed over the 10 AD setups with 10 seeds per setup (i.e. overall 100 runs) at various ratios.

Data Set	% Labeled Train Set	Soft DSVDD	One-Class DSVDD	Soft SS-DSVDD	One-Class SS-DSVDD	Supervised Classifier
MNIST	0%	94.1±1.0	**95.0±0.8**	94.1±1.0	**95.0±0.8**	
	1%			94.1±1.1	**96.2±0.8**	90.1±4.0
	5%			94.7±1.2	**96.8±0.9**	92.6±3.4
	10%			95.0±1.2	**97.0±1.0**	93.0±3.2
	20%			95.3±1.2	**97.1±1.0**	93.8±2.6
Fashion-MNIST	0%	91.1±0.5	**91.6±0.5**	91.1±0.5	**91.6±0.5**	
	1%			91.3±0.7	**92.8±1.0**	78.6±10.9
	5%			91.7±1.0	**93.1±1.2**	84.1±7.2
	10%			91.9±1.2	**93.2±1.3**	85.4±5.8
	20%			92.0±1.3	**93.3±1.3**	87.5±4.9
CIFAR-10	0%	63.9±1.7	**64.7±1.7**	63.9±1.7	**64.7±1.7**	
	1%			64.6±1.9	**69.6±3.1**	55.6±5.0
	5%			68.3±2.3	**73.6±4.2**	63.5±8.0
	10%			69.4±2.8	**75.3±4.6**	67.7±9.6
	20%			71.7±3.0	77.0±4.4	**80.5±5.9**

Table 2. Experimental results when gradually polluting the unlabeled part of the training set with anomalies. We report the average AUC with standard deviation computed over the 10 AD setups with 10 seeds per setup (i.e. overall 100 runs) at various pollution ratios.

Data Set	% Pollution Train Set	Soft DSVDD	One-Class DSVDD	Soft SS-DSVDD	One-Class SS-DSVDD	Supervised Classifier
MNIST	0%	94.1±1.0	95.0±0.8	94.7±1.2	**96.8±0.9**	92.6±3.4
	1%	90.3±1.6	94.4±0.8	89.9±3.9	**96.0±0.9**	90.3±4.2
	5%	86.6±1.6	92.1±1.2	85.9±2.3	**94.4±1.3**	85.5±5.4
	10%	83.2±1.6	89.6±1.4	82.7±1.8	**92.4±1.4**	82.4±6.4
	20%	79.3±1.5	85.5±1.5	78.9±1.5	**88.9±1.5**	78.2±7.6
Fashion-MNIST	0%	91.1±0.5	91.6±0.5	91.7±1.0	**93.1±1.2**	84.1±7.2
	1%	88.5±1.1	89.4±0.8	84.7±7.1	**91.3±1.0**	79.1±9.6
	5%	83.0±1.0	84.5±0.8	82.6±2.5	**87.3±1.8**	71.9±11.4
	10%	78.6±1.1	80.5±1.1	79.5±2.0	**83.8±2.8**	69.8±11.8
	20%	74.5±1.4	76.1±1.2	75.7±1.6	**79.8±3.3**	64.5±12.6
CIFAR-10	0%	63.8±1.5	64.4±1.7	68.3±2.3	**73.6±4.2**	63.5±8.0
	1%	62.7±1.7	63.9±1.5	70.7±6.3	**70.8±3.8**	62.9±7.3
	5%	61.5±1.6	62.5±1.4	67.6±4.3	**69.4±3.8**	62.2±8.2
	10%	60.6±1.6	61.7±1.7	64.6±3.1	**69.6±4.6**	60.6±8.3
	20%	59.0±1.6	59.8±1.7	61.2±2.1	**68.4±5.0**	58.5±6.7

Table 3. Experimental results when gradually increasing the number of known anomaly classes. We report the average AUC with standard deviation computed over the 10 AD setups with 10 seeds per setup (i.e. overall 100 runs) at various numbers of known anomaly classes.

	# Known	Soft	One-Class	Soft	One-Class	Supervised
Data Set	Classes	DSVDD	DSVDD	SS-DSVDD	SS-DSVDD	Classifier
MNIST	0	83.2±1.6	**89.6±1.4**	83.2±1.6	**89.6±1.4**	
	1			82.7±1.8	**92.4±1.4**	82.4±6.4
	2			83.9±1.6	**93.7±1.4**	89.7±3.0
	3			84.7±1.6	**94.6±1.3**	92.6±2.5
	5			85.4±1.4	95.5±1.2	**96.2±1.0**
Fashion-MNIST	0	78.6±1.1	**80.5±1.1**	78.6±1.1	**80.5±1.1**	
	1			79.5±2.0	**83.8±2.8**	69.8±11.8
	2			81.2±1.9	**86.4±2.6**	78.6±8.2
	3			82.3±1.7	**88.7±2.1**	83.9±6.1
	5			83.5±1.4	**91.7±1.4**	89.0±3.4
CIFAR-10	0	60.6±1.6	**61.7±1.7**	60.6±1.6	**61.7±1.7**	
	1			64.6±3.1	**69.6±4.6**	60.6±8.3
	2			64.9±2.5	**71.7±4.6**	61.0±6.6
	3			65.0±2.3	**72.1±4.4**	62.7±6.8
	5			65.2±1.8	**73.0±4.5**	60.9±4.6