Literature Review

August 16, 2021

Abstract

In this document, we collect related work. We divide papers in *surveys* and *novel AD-methods*. For each paper, we record a link, and a small explanation of why the paper is relevant. Almost any single paper on AD *will* be relevant here. We care about how hyperparameters are selected for benchmarking, which is something that happens all the time.

1 Papers proposing a novel AD-method

1.0.1 2020 - Courville - Semantic AD

- Clickable link
- Reference [1]
- Relevant quotes for categorization
  - *For ODIN, it is unclear how to choose the hyperparameters for temperature scaling and the weight for adversarial perturbation without assuming access to anomalous examples, an assumption we consider unrealistic in most practical settings. We fix $T = 1000$, $\epsilon = 5e\text{-}5$ for all experiments, following the most common setting.*
  - *We note that ODIN, with fixed hyperparameter settings across all experiments, continues to outperform MSP most of the time.*
  - *Our base network for all CIFAR-10 experiments is a Wide ResNet (Zagoruyko and Komodakis, 2016) with 28 convolutional layers and a widening factor of 10 (WRN-28-10) with the recommended dropout rate of 0.3. Following Zagoruyko and Komodakis (2016), we train for 200 epochs, with an initial learning rate of 0.1 which is scaled down by 5 at the 60th, 120th, and 160th epochs, using stochastic gradient descent with Nesterov's momentum at 0.9*
  - *For experiments with the proposed subsets of IMAGENET, we replicate the architecture we use for STL-10, but add a downsampling average pooling layer after the first convolution on the images. We do not use dropout, and use a batch size of 64, train for 200 epochs; otherwise all other details follow the settings for STL-10*

- *For adding rotation-prediction as an auxiliary task, all we do is append an extra linear layer alongside the one that is responsible for object recognition. λ is tuned to 0.5 for CIFAR-10, 1.0 for STL-10, and a mix of 0.5 and 1.0 for IMAGENET*

- Almost all hyperparameters are kept at "default" hyperparameters, with the only exception being the hyperparameter λ, relevant for one auxiliary task only.

1.0.2 2020 - Datta - AD for Timber Trades

- Clickable link
- Reference [2]
- Relevant quotes for categorization
  - *We set hyper-parameters $\alpha = 0.1$ and $\beta = 0.1$. k is set to 2 for dataset CE and 1 for others.*
  - *We used the authors original implementation for our experiments*
  - For their own method, the authors report some plots that show the impact of various hyperparameters. It is unclear from the paper which exact parametrization is used to generate the results in the final comparison, however. Additionally, it should be noted that the authors work in an inductive setting, i.e. there is a train and test set.
- Default hyperparameters for the competition.

1.0.3 2020 - Vercruyssen - Transfer Learning AD Localized and Unsupervised Instance Selection

- Clickable link
- Reference [3]
- Relevant quotes for categorization
  - *Hyperparameter tuning using cross-validation is impossible because there are no labels in the target domain and the distribution of the source data is different (Pan et al. 2011).*
  - *We simply use the baselines with the hyperparameters recommended in the original papers or in comparative studies. LOCIT has two hyperparameters. We set the neighborhood size $\psi = 20$ and $k = 10$ in SSKNNO.*
- Hyperparameter selection
  - Default hyperparameters. The authors indicate that they have to, given the fact that in their setup there are no labels on which to do crossvalidation. This closely aligns to what one would expect when doing AD in practice.

1.0.4 2019 - Gutflaish - Temporal AD

- Clickable link
- Reference [4]
- Relevant quotes for categorization
  - *In each of the first four baselines, we used the default parameters as recommended by the authors.*
- Hyperparameter selection
  - Default hyperparameters for baselines (i.e. methods with which the proposal is compared)

1.0.5 2019 - Sheng - Multi-view AD

- Clickable link
- Reference [5]
- Relevant quotes for categorization
  - *We compare the proposed MUVAD-QPR and MUVAD-FSR approaches with OCSVM (Schölkopf et al. 2001), HOAD (Gao et al. 2011), CC (Liu and Lam 2012), MLRA (Li, Shao, and Fu 2015), DMOD (Zhao and Fu 2015) and CRMOD (Zhao et al. 2018). CRMOD is the extended version of DMOD.*
  - *As for MUVAD-QPR, we use t = 7,N0 = 0.9N,λ = -2000 as default parameters. As for MUVAD-FSR, we use t = 7, γ = 2000 as default parameters.*
- Hyperparameter selection
  - Default hyperparameters, for MUVAD-QPR and MUVAD-FSR, the proposed solution.
  - Details about the hyperparameter selection of the competitors is absent, although the way it is phrased seems to suggest the default *implementations* are used, and therefore default hyperparameters as well. In any case, there is no mention of any kind of tuning in the experiments.

1.0.6 2019 - Feremans & Vercruyssen - Pattern-based AD in time-series

- Clickable link
- Reference [6]
- Relevant quotes for categorization
  - *The parameters of Fpof and Mifpod are chosen by an oracle that knows the optimal settings for each dataset.*
- Hyperparameter selection
  - Optimal hyperparameters are used for all competitors that actually require hyperparameters to be set.
  - The authors' own proposal relies on reasonable defaults.

1.0.7 2019 - Gopalan - PIDForest

- Clickable link
- Reference [7]
- Relevant quotes for categorization
  - *Except for RRCF, we run each algorithm with the default hyperparameter setting as varying the hyperparameters from their default values did not change the results significantly.*
  - *For RRCF, we use 500 trees instead of the default 100 since it yielded significantly better performance*
  - *For PIDForest, we fix the hyperparameters of depth to 10, number of trees to 50, and the number of samples used to build each tree to 100*
- Hyperparameter selection
  - Defaults for almost all algorithms, including own proposal.
  - For RRCF, a non-default value for a hyperparameter was used to yield better results. No details on how this was obtained.

1.0.8 2019 - Babaei - PLOF

- Clickable link
- Reference [8]
- Relevant quotes for categorization
  -
- Hyperparameter selection
  - Unclear; *probably* "best-average"

1.0.9 2018 - Pang - SparseModelling OD

- Clickable link
- Reference [9]
- Relevant quotes for categorization
  - In all our experiments, CINFO uses a = 1.732 (i.e., the upper bound for false positives in η is 25%) and m = 30 2; and the number of subsamples l and subsampling size $|M|$ for LeSiNN and iForest are set as the recommended settings of their authors.
- Hyperparameter selection
  - Own method: fixed values reported, details missing on how they are chosen.
  - Other methods: mostly defaults and once "best-average".

1.0.10 2018 - Chalapathy - OC-NN

- Clickable link
- Reference [10]
- Relevant quotes for categorization
  - *The model parameters of shallow baseline methods are used as per implementation in Ruff et al. [2018]. Shallow Baselines (i) Kernel OC-SVM/SVDD with Gaussian kernel. We select the inverse length scale* γ *from* $\gamma \in 2-10, 2-9, ..., 2-1$ *via grid search using the performance on a small holdout set (10) % of randomly drawn test samples). We run all experiments for* $\nu = 0.1$ *and report the better result. (ii) Kernel density estimation (KDE). We select the bandwidth h of the Gaussian kernel from h ∈ 20.5 , 21 , ..., 25 via 5-fold cross-validation using the log-likelihood score. (iii) For the Isolation Forest (IF) we set the number of trees to t = 100 and the sub-sampling size to* $\phi = 256$*, as recommended in the original work Ruff et al. [2018].*
  - *We compare OC-NN models to four deep approaches described Section 4.1. We choose to train DCAE using the Mean square error (MSE) loss since our experiments are on image data. For the DCAE encoder, we employ the same network architectures as we use for Deep SVDD, RCAE, and OC-NN models. The decoder is then constructed symmetrically, where we substitute max-pooling with upsampling. For AnoGAN we follow the implementation as per Radford et al. [2015] and set the latent space dimensionality to 256. For we Deep SVDD, follow the implementation as per Ruff et al. [2018] and employ a two phase learning rate schedule (searching + fine tuning) with initial learning rate* $\eta = 10-4$ *and subsequently* $\eta = 10-5$*. For DCAE*

we train 250 + 100 epochs, for Deep SVDD 150 + 100. Leaky ReLU activations are used with leakiness $\alpha = 0.1$. For RCAE we train the autoencoder using the robust loss and follow the parameter settings as per formulation in Chalapathy et al. [2017].

- W.r.t. own proposal: *A feed-forward neural network consisting of single hidden layer, with linear activation functions* ***produced the best results****, as per Equation 3. The* ***optimal value of parameter*** $\nu \in [0, 1]$ *which is equivalent to the percentage of anomalies for each data set, is set according to respective outlier proportions.*

- Hyperparameter selection
  - Competitors: mostly default hyperparameters, sometimes optimal hyperparameters via grid search and crossvalidation.
  - Own proposal: optimal hyperparameters.

1.0.11 2018 - Golan - AD with deep geometric transformations

- Clickable link
- Reference [11]
- Relevant quotes for categorization
  - *It is very important to note that in both these variants of OC-SVM, we provide the OC-SVM with an* ***unfair significant advantage by optimizing its hyperparameters in hindsight****; i.e., the OC-SVM hyperparameters (ν and γ) were* ***optimized to maximize AUROC and taken to be the best performing values*** *among those in the parameter grid:* $\nu \in \{0.1, 0.2, ..., 0.9\}, \gamma \in \{2 - 7, 2 - 6, ..., 22\}$. *Note that the hyperparameter optimization procedure has been provided with a two-class classification problem*
  - *The convolutional autoencoder is chosen to have a similar architecture to that of DCGAN [26], where the encoder is adapted from the discriminator, and the decoder is adapted from the generator*
  - *The experimental setup used by the authors is identical to ours, allowing us to report their published results as they are, on CIFAR-10.*
  - *The chosen architecture is the same as that of the encoder part in the convolutional autoencoder used by CAE- OC-SVM, with ReLU activations in the encoding layer.*
  - *The architecture of the autoencoder we used is similar to that of the convolutional autoencoder from the CAE-OC-SVM experiment, but with linear activation in the representation layer. The estimation network is inspired by the one in the original DAGMM paper.*

  - *In our experiments, for the generative model of the ADGAN we incorporated the same architecture used by the authors of the original paper, namely, the original DCGAN architecture [26]. As described, ADGAN requires only a trained generator.*
  - *Our model is implemented using the state-of-the-art Wide Residual Network (WRN) model [40]. The parameters for the depth and width of the model for all 32x32 datasets were chosen to be 10 and 4, respectively, and for the CatsVsDogs dataset (64x64), 16 and 8, respectively. These hyperparameters were selected prior to conducting any experiment, and were fixed for all runs.3 We used the Adam [20] optimizer with default hyperparameters. Batch size for all methods was set to 128. The number of epochs was set to 200 on all benchmark models, except for training the GAN in ADGAN for which it was set to 100 and produced superior results. We trained the WRN for* $\lceil 200/|T| \rceil$ *epochs on the self-labeled set ST , to obtain approximately the same number of parameter updates as would have been performed had we trained on S for 200 epochs.*

- Hyperparameter selection
  - For OC-SVM-based competitors (there are 2 of those), optimal hyperparameters are used.
  - Reasonable defaults for all other methods. Sometimes taken from literature/implementations, but not always.

1.0.12 2018 - Xu - VAE for AD

- Clickable link
- Reference [12]
- Relevant quotes for categorization
  - We set the window size W to be 120, which spans 2 hours in our datasets.
  - We set the latent dimension K to be 3 for B and C, since the 3-d dimensional space can be easily visualized for analysis and luckily K = 3 works well empirically for for B and C. As for A, we found 3 is too small, so we empirically increase K to 8. These empirical choices of K are proven to be quite good on **testing set**, as will be shown in Fig 10.
  - The hidden layers of $p_\phi(z|x)$ and $p_\theta(z|x)$ are both chosen as two ReLU layers, each with 100 units, which makes the variational and generative network have equal size. **We did not carry out exhaustive search on the structure of hidden networks.**

  - **Other hyper-parameters are also chosen empirically**. We use 10^{-4} as ϵ of the std layer. We use 0.01 as the injection ratio λ. We use 10 as the MCMC iteration count M, and use 1024 as the sampling number L of Monte Carlo integration. We use 256 as the batch size for training, and run for 250 epochs. We use Adam optimizer [15], with an initial learning rate of 10^{-3}. We discount the learning rate by 0.75 after every 10 epochs. We apply L2 regularization to the hidden layers, with a coefficient of 10^{-3}. We clip the gradients by norm, with a limit of 10.0.

- Hyperparameter selection
  - Own method: unclear, *probably* combination of defaults and some manual tuning ("We did not carry out exhaustive search on the structure of hidden networks").
  - Other methods: unclear, *probably* defaults.

1.0.13 2016 - Iwata - Multi-View AD

- Clickable link
- Reference [13]
- Relevant quotes for categorization:
  - For PCCA, we used the proposed model in which **the number of latent vectors was fixed at one for every instance**.
  - HOAD requires to select an appropriate hyperparameter value for controlling the constraints whereby different views of the same instance are embedded close together. We ran HOAD with different hyperparameter settings 0.1, 1, 10, 100, and **show the results that achieved the highest performance for each data set.**
  - For CC, first we clustered instances for each view using spectral clustering. **We set the number of clusters at 20**, which achieved a good performance in preliminary experiments.
  - For OCSVM, multiple views are concatenated in a single vector, then use it for the input. We used Gaussian kernel. In the proposed model, we used $\gamma = 1$, $a = 1$, **and** $b = 1$ **for all experiments**. **The number of iterations for the Gibbs sampling was 500**, and the anomaly score was calculated by averaging over the multiple samples.
- Hyperparameter selection
  - Peak performance for one method
  - Defaults for others
  - For own method also defaults, but no explanation how these are chosen

1.0.14 2016 - Hsiao - Pareto AD

- Clickable link
- Reference [14]
- Relevant quotes for categorization
  - *The median and* ***best AUCs (over all choices of weights selected by grid search) are shown for the other four methods.*** *PDA outperforms all of the other methods, even for the best weights, which are not known in advance.*
  - *We compare the PDA method with four other nearest neighbor-based single-criterion anomaly detection algorithms mentioned in Section 2. For these methods, we use linear combinations of the criteria with different weights* ***selected by grid search*** *to compare performance with PDA.*
  - *A grid of six points between 0 and 1 in each criterion, corresponding to 64 = 1296 different sets of weights, is used to select linear combinations for the single-criterion methods. Note that PDA is the best performer,* ***outperforming even the best linear combination.***
- Hyperparameter selection
  - Optimal hyperparameters for competitors. The authors run the competition over a grid of hyperparameters, and compare their own proposal with the best performing setting of the competition.
  - Defaults hyperparameters for own proposal.

1.0.15 2016 - Pevny - LODA

- Clickable link
- Reference [15]
- Uses defaults, and in one experiment does a poorly explained train/test split.

1.0.16 2015 - Paulheim - ALSO

- Clickable link
- Reference [16]
- Competitors + Own method: Defaults

1.0.17 2014 - Kriegel & Zimek - KDEOS

- Clickable link
- Reference [17]
- Relevant quotes for categorization:
  - *Table 1:* ***Best performance*** *of different algorithms.*
  - *The best results and the k for the best result are given in Table 1.*
  - *Our proposed method produces much more stable results, in particular when choosing a large enough range of ks.* ***Besides offering the best performance of the evaluated algorithms****, the parameter k is also much easier to choose – it just needs to be large enough for the kernel density estimation to yield meaningful results.*
  - *Figure 2: Performance over different values of k.*
- Hyperparameter selection
  - The impact of the influential hyperparameter k is visualized and discussed.
  - Table 1 reports peak performance and used to make the case that the proposed method outperforms the competition.

1.0.18 2013 - Lu - Student-t for mixed-type AD

- Clickable link
- Reference [18]
- Relevant quotes for categorization:
  - *For both LOADED and RELOADED, we tried popular settings of the model parameters (correlation threshold = [0.1, 0.2, 0.3, 0.5, 0.8, 1]; frequency threshold = [0, 10, 20];* τ *= [1,2,3,5]), and* ***reported the best results for each dataset based on true anomaly labels****.*
  - *For the other three approaches, the parameters were selected based on 10-fold cross validations*
- Hyperparameter selection
  - Peak performance; results of (quasi) optimal hyperparameters per dataset are reported.
  - N.B. Cross-validation to select appropriate hyperparameters still requires all labels to be known a priori, and thus would still be impossible to do in a practical setting.

1.0.19 2013 - Jannsens - SOS

- Clickable link
- Reference [19]
- Relevant quotes for categorization (all from pp. 91):
  - ***The figure reveals that SOS has a superior performance on twelve data sets.***
  - *For completeness, the AUC performances of SOS, KNNDD, LOF, LOCI, and LSOD on the 47 real-world one-class data sets are stated in Table 4.2 for various parameter settings.*
  - *In addition to* ***the maximum achieved AUC performance as is also shown in Figure 4.16****, Table 4.2 also shows the performances for other parameter values.*
- Hyperparameter selection
  - To draw conclusions and to compare algorithms: peak performance.
  - For completeness, results with other configurations are reported in an extra Table.

1.0.20 2013 - Nguyen - CMI

- Clickable link
- Reference [20]
- Competitors: Defaults
- Own method: Unclear; *probably* peak performance (i.e. perfect hyperparameters)
- Again, an illustration that "using defaults" is just inconsistent.

1.0.21 2013 - Sugiyama - Rapid DBOD via Sampling

- Clickable link
- Reference [21]
- Defaults for everyone, including own method. Own default is chosen "to be consistent" with a semantically similar parameter of a competitor.

1.0.22 2012 - Kriegel Zimek - COP

- Clickable link
- Reference [22]
- Extremely unclear: almost no mention of how parameters where set. Both for own method, as well as competitor.

1.0.23 2012 - Keller - HiCS

- Clickable link and Clickable link
- Reference [23]
- Relevant quotes for categorization
  - Experiment 01: *In total, 21 synthetic datasets were generated: 3 datasets for each dimensionality in [10, 20, 30, 40, 50, 75, 100]. We performed several algorithmic configurations on all these datasets.*
  - Experiment 01: *In our comprehensive quality experiment (cf. Fig. 4), we have noticed a high sensitivity w.r.t. parametrization for our competitors. For RIS and Enclus in particular, we have observed that finding good parameter settings is difficult. Therefore we had run the whole experiment with a large number of configurations for these two algorithms.* ***We have shown only the best values in the previous graphs.***
  - Experiment 02: *The following experiments were performed with each best algorithm configuration from the experiment on synthetic data.*
- Hyperparameter selection
  - Best-default hyperparameters in experiment 01. Slightly ambiguous phrasing, it could be that experiment 01 reports peak performance instead, i.e. per dataset, report the result of the best configuration. However, the fact that experiment 02 uses "each best algorithm configuration" from experiment 01 suggests that a single configuration per algorithm is used across experiment 01 and experiment 02.
  - Those best-default hyperparameters hyperparameters found and used in experiment 01 are assumed to be "default" hyperparameters in experiment 02.

1.0.24 2011 - Xiong - FGM

- Clickable link
- Reference [24]
- Fixed hyperparameters for all, defaults chosen via BIC ("*as it was suggested by BIC searches*").

1.0.25 2010 - Muller - OUTRES

- Clickable link or Clickable link
- Reference [25]
- Absolutely zero information about hyperparameters.

1.0.26 2009 - Zhao - k-LPE

- Clickable link
- Reference [26]
- Relevant quotes for categorization
  - *Finally we iterated over different c to obtain the best (in terms of AUC) ROC curve and it turns out to be c = 1.5.*
  - *To test the sensitivity of K-LPE to parameter changes, we first run K-LPE on the benchmark ar- tificial data-set Banana [19] with K varying from 2 to 12.*
- Hyperparameter selection
  - This paper uses four datasets, but only on one of those there is a comparison with OCSVM.
  - For OCSVM, peak-performance is reported.
  - For own proposal, results for multiple K are reported.

1.0.27 2009 - Kriegel - LooP

- Clickable link
- Reference [27]
- Report all results, across hyperparameters.

1.0.28 2009 - Zimek & Kriegel - SOD

- Clickable link
- Reference [28]
- Absolutely zero information on hyperparameters reported.

1.0.29 2008 - Liu - iForest

- Clickable link
- Reference [29]
- Competitors: Defaults
- Own method: Unclear; presumably best average.

1.0.30 2008 - Deng - CBLOF

- Clickable link
- Reference [30]
- Competitors: Fixed, unclear how values are chosen. *Probably* best-average.
- Own method: Fixed, unclear how values are chosen. *Probably* best-average.

1.0.31 2008 - Kriegel & Zimek - ABOD

- Clickable link
- Reference [31]
- Competitors: Just LOF, multiple values tried, unclear what is being reported in the figures. Presumably peak-performance.
- Own method: No parameters, except for the subsampling version, there a default is used.

1.0.32 2007 - Latecki - LDF

- Clickable link
- Reference [32]
- Competitors + Own: No clear methodology, different values for different datasets, presumably peak performance. However, LoCI was ran with default parameters.

1.0.33 2006 - Pei - Reference-based Outlier Detection (RBOD

- Clickable link
- Reference [33]
- Competitors + Own: In synthetic data fixed parameters, unclear whether "default", or "best average".
- In subsequent experiments, parameter values are reported, but not clear why. Most likely "peak performance".

1.0.34 2004 - Hautamaki - ODIN

- Clickable link
- Reference [34]
- Relevant quotes for categorization
  - *Table 2 summarizes* ***parameters that give minimum error rate for each algorithm****.*
  - *Table 2 shows that* ***optimal parameters for each dataset vary greatly****, this leads to a problem of how to find the optimal parameter combination in the 2d parameter space.*
- Hyperparameter selection
  - Optimal hyperparameters for competitors and own proposal. N.B. Hyperparameters are chosen on a per-dataset basis.

1.0.35 2003 - Papadimitriou - LOCI

- Clickable link
- Reference [35]
- Competitors + Own: Once again very unclear, but appears to be peak performance.

1.0.36 2001 - Scholkopt - OCSVM

- Clickable link
- Reference [36, 37]
- Own method: uses defaults
- Competitors: No competitors in paper...

1.0.37 2000 - Eskin - GaussianUniformMixture

- Clickable link
- Reference [38]
- Competitors: Defaults
- Own method: Defaults, claimed to be unoptimized.

1.0.38 1994 - Tarassenko - Probabilistic Resource Allocating Network for Novelty Detection

- Clickable link
- Reference [39]
- Own method: suggests defaults, unclear how/why except "they are not crucial"

2 Surveys

2.0.1 2021 - Ruff, Dietterich et al. - A Unifying Review of Deep and Shallow Anomaly Detection

- Clickable link
- Reference [40]
- Some defaults, some neural net parameters tuned on a "small holdout set", some neural nets are pre-trained.

2.0.2 2021 - Marques & Zimek - Internal Evaluation of AD

- Clickable link
- Reference [41]
-

2.0.3 2019 - Gu - Statistical Analysis of Nearest-Neighbor AD

- Clickable link
- Reference [42]
- Default hyperparameters
- They (surprisingly) seem to suggest that nearest-neighbor methods are in fact *robust* wrt the choice of hyperparameters
  - *The discussion on the robustness of distance-based methods to the choice of hyperparameter k can be found at [27] [28].*
  - N.b. this quote follows directly after they summarize all their default settings, which is why it kind of reads as if "k does not matter much".
  - Reference [28] is in fact Campos, which makes the exact opposite point that these methods are not robust against the choice of k?

2.0.4 2016 - Goldstein - Survey

- Clickable link
- Reference [43]
- Reports AVG+STD when varying k in NN-based approaches.
- For other parameters, mostly pick "defaults from authors", unless clearly problematic $\Rightarrow$ very ad-hoc in the end.

- However, for some algorithms such as e.g. SVM, they do report again an average across multiple choices for the same parameters, which assumes "completely random defaults".
- On hyperparameters etc, they make the following points.
  - *In most publications, researchers often fix k to a predefined setting or choose "a good k" depending on a favorite outcome. We believe that the latter is not a fair evaluation, because it somehow involves using the test data (the labels) for training*
  - *In our evaluation, we decided to evaluate many different k's between 10 and 50 and finally report the averaged AUC as well as the standard deviation.*
  - *This procedure basically corresponds to a random-k-picking strategy within the given interval, which is often used in practice when k is chosen arbitrarily.*

2.0.5 2016 - Campos - Survey

- Clickable link
- Reference [44]
- *FastABOD uses the kernel trick, and thus an appropriate kernel function must be chosen (we use the default polynomial kernel of degree 2). KDEOS performance benefits from choosing an intrinsic dimensionality and a kernel bandwidth multiplier h. LDF has a comparable kernel bandwidth multiplier constant h, and an additional score scaling constant c.* ***We do not vary these default settings*** *(we use $h = 1$, $c = 0.1$), though, as we evaluate all algorithms on the same terms,* ***varying only the neighborhood size*** *[i.e. k]. For many datasets, there may be better results obtainable with either algorithm by further exploring the parameter space.*
- *It is thus difficult to ascertain the extent to which newly-proposed outlier detection methods improve over established methods*
- Thus, the authors report results across k, but other parameters remain fixed at defaults. *For many datasets, there may be better results obtainable with either algorithm by further exploring the parameter space*
- For as far k is concerned they do two things: *To compare the methods according to their quality scores we will consider initially (1) the average performance over the range of given values of k (representing an expected performance if the users have no prior knowledge about k), and (2)* ***the best-case performance, selecting the k for which the performance of a method on a dataset is maximal (representing the optimistic case where the optimal value of k for a method is known in advance)***

  - So for k, the most influential parameter **this is peak performance** and basically "random default k".
  - Given the above, we classify this paper under "peak performance".

- Interestingly, the authors do make the point that choosing hyperparameters is very important, and that simply relying on defaults can lead to misleading results;

  - *We also show that testing broad ranges of parameter values is crucial when evaluating outlier methods, in order to avoid misleading experimental outcomes..*
  - *These findings also shed some light as to how the evaluation of new methods could be performed. It is always possible to find cases (specific parameter settings for specific datasets) where one particular method outperforms its competitors. As was demon- strated here, best practice dictates that the behavior of outlier detection methods be studied across a range of parameter settings, as the results for different parameter val- ues can vary widely. Even if the methods to be compared share a seemingly analogous parameter (such as a neighborhood size k), setting it to the same values for all meth- ods may still not allow for a direct comparison. As indicated in Fig. 1, the methods may depend on the parameter in different ways, and reach their peak performances for different choices of a seemingly identical parameter such as neighborhood size. Surveying the research literature would suggest that best practice is not always fol- lowed. There are many publications in which methods are compared and conclusions are made based on only a single, arbitrary choice of an important parameter (see the work of Müller et al. (2011, 2012), Liu et al. (2012), Keller et al. (2012), Ting et al. (2013) for some recent examples published at high quality venues).*
  - *However, by picking appropriate parameter values, one may cast any of the methods tested in a favorable light, which emphasizes the importance of systematic testing across a range of parameter values*

2.0.6 2015 - Emmott & Dietterich - Meta-Analysis AD Problem

- Clickable link
- Reference [45]
- This is a follow up of their previous survey
- They make the point that experimental design has a large influence on subsequent conclusions: this is a point we also want to make, but then specifically for the topic of hyperparameter selection.
- Again, they make the point that methodologies are not fixed enough; their focus is specifically the issue of benchmark datasets.

- *Instead we aim to highlight the common pitfalls associated with evaluating success in this field and the difficulty in measuring progress in the field.*
- For each dataset, an effort was done to parametrize as good as possible. Sometimes following specific methods from the literature, sometimes based on own experience, sometimes based on defaults.

2.0.7 2014 - Kriegel & Zimek - LOD reconsidered

- Clickable link
- Reference [46]
- Unclear, this is specifically about spatial data and (in the limited time available) I cannot really understand how they modify existing algorithms and what is going on with the hyperparameters.

2.0.8 2013 - Emmott & Dietterich - Systematic Construction of AD Benchmarks

- Clickable link
- Reference [47]
- They make the explicit point that many benchmark datasets are needed for meaningful results.
- They employ somewhat different strategies for each method but: "*in all cases, we made a good faith effort to maximize the performance of all methods*". OCSVM for instance is tuned via cross-validation, but for LOF/iForest they just use a rule of thumb (parameters set relative to number of instances) that yielded good results.
- Ultimately, this boils down to reporting peak performance, and in some cases the tuning did happen in a sound way.

References

[1] Faruk Ahmed and Aaron Courville. Detecting semantic anomalies. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3154–3162, Apr. 2020.

[2] Debanjan Datta, M. Raihanul Islam, Nathan Self, Amelia Meadows, John Simeone, Willow Outhwaite, Chen Hin Keong, Amy Smith, Linda Walker, and Naren Ramakrishnan. Detecting suspicious timber trades. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(08):13248–13254, Apr. 2020.

[3] Vincent Vercruyssen, Wannes Meert, and Jesse Davis. Transfer learning for anomaly detection through localized and unsupervised instance selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6054–6061, Apr. 2020.

[4] Eyal Gutflaish, Aryeh Kontorovich, Sivan Sabato, Ofer Biller, and Oded Sofer. Temporal anomaly detection: Calibrating the surprise. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3755–3762, Jul. 2019.

[5] Xiang-Rong Sheng, De-Chuan Zhan, Su Lu, and Yuan Jiang. Multi-view anomaly detection: Neighborhood in locality matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4894–4901, Jul. 2019.

[6] Len Feremans, Vincent Vercruyssen, Boris Cule, Wannes Meert, and Bart Goethals. Pattern-based anomaly detection in mixed-type time series. In Ulf Brefeld, Élisa Fromont, Andreas Hotho, Arno J. Knobbe, Marloes H. Maathuis, and Céline Robardet, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part I*, volume 11906 of *Lecture Notes in Computer Science*, pages 240–256. Springer, 2019.

[7] Parikshit Gopalan, Vatsal Sharan, and Udi Wieder. Pidforest: Anomaly detection via partial identification. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[8] Kasra Babaei, ZhiYuan Chen, and Tomas Maul. Detecting point outliers using prune-based outlier factor (plof). *arXiv preprint arXiv:1911.01654*, 2019.

[9] Guansong Pang, Longbing Cao, Ling Chen, Defu Lian, and Huan Liu. Sparse modeling-based sequential ensemble learning for effective outlier detection in high-dimensional numeric data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[10] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

[11] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[12] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pages 187–196, 2018.

[13] Tomoharu Iwata and Makoto Yamada. Multi-view anomaly detection via robust probabilistic latent variable models. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[14] Ko-jen Hsiao, Kevin Xu, Jeff Calder, and Alfred Hero. Multi-criteria anomaly detection using pareto depth analysis. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

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

[16] Heiko Paulheim and Robert Meusel. A decomposition of the outlier detection problem into a set of supervised learning problems. *Machine Learning*, 100(2-3):509–531, 2015.

[17] Erich Schubert, Arthur Zimek, and Hans-Peter Kriegel. Generalized outlier detection with flexible kernel density estimates. In Mohammed Javeed Zaki, Zoran Obradovic, Pang-Ning Tan, Arindam Banerjee, Chandrika Kamath, and Srinivasan Parthasarathy, editors, *Proceedings of the 2014 SIAM International Conference on Data Mining, Philadelphia, Pennsylvania, USA, April 24-26, 2014*, pages 542–550. SIAM, 2014.

[18] Yen-Cheng Lu, Feng Chen, Yang Chen, and Chang-Tien Lu. A generalized student-t based approach to mixed-type anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 27(1), Jun. 2013.

[19] J.H.M. Janssens. *Outlier selection and one-class classification*. PhD thesis, 2013. Series: TiCC Ph.D. Series Volume: 27.

[20] Hoang Vu Nguyen, Emmanuel Müller, Jilles Vreeken, Fabian Keller, and Klemens Böhm. Cmi: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In *Proceedings of the 2013*

SIAM International Conference on Data Mining, pages 198–206. SIAM, 2013.

[21] Mahito Sugiyama and Karsten Borgwardt. Rapid distance-based outlier detection via sampling. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[22] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Outlier detection in arbitrarily oriented subspaces. In Mohammed Javeed Zaki, Arno Siebes, Jeffrey Xu Yu, Bart Goethals, Geoffrey I. Webb, and Xindong Wu, editors, *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pages 379–388. IEEE Computer Society, 2012.

[23] Fabian Keller, Emmanuel Muller, and Klemens Bohm. Hics: high contrast subspaces for density-based outlier ranking. In *IEEE 28th Intl. Conference on Data Engineering*, pages 1037–1048. IEEE, 2012.

[24] Liang Xiong, Barnabás Póczos, and Jeff Schneider. Group anomaly detection using flexible genre models. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[25] Emmanuel Müller, Matthias Schiffer, and Thomas Seidl. Adaptive outlierness for subspace outlier ranking. In Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An, editors, *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1629–1632. ACM, 2010.

[26] Manqi Zhao and Venkatesh Saligrama. Anomaly detection with score functions based on nearest neighbor graphs. *arXiv preprint arXiv:0910.5461*, 2009.

[27] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Loop: Local outlier probabilities. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 1649–1652, New York, NY, USA, 2009. Association for Computing Machinery.

[28] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In Thanaruk Theeramunkong, Boonserm Kijsirikul, Nick Cercone, and Tu Bao Ho, editors, *Advances in Knowledge Discovery and Data Mining, 13th Pacific-Asia Conference, PAKDD 2009, Bangkok, Thailand, April 27-30, 2009, Proceedings*, volume 5476 of *Lecture Notes in Computer Science*, pages 831–838. Springer, 2009.

[29] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE Intl. Conference on Data Mining*, pages 413–422. IEEE, 2008.

[30] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognit. Lett.*, 24(9-10):1641–1650, 2003.

[31] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. Angle-based outlier detection in high-dimensional data. In Ying Li, Bing Liu, and Sunita Sarawagi, editors, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 444–452. ACM, 2008.

[32] Longin Jan Latecki, Aleksandar Lazarevic, and Dragoljub Pokrajac. Outlier detection with kernel density functions. In Petra Perner, editor, *Machine Learning and Data Mining in Pattern Recognition, 5th International Conference, MLDM 2007, Leipzig, Germany, July 18-20, 2007, Proceedings*, volume 4571 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2007.

[33] Yaling Pei, Osmar R. Zaïane, and Yong Gao. An efficient reference-based approach to outlier detection in large datasets. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China*, pages 478–487. IEEE Computer Society, 2006.

[34] Ville Hautamäki, Ismo Kärkkäinen, and Pasi Fränti. Outlier detection using k-nearest neighbour graph. In *17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, August 23-26, 2004*, pages 430–433. IEEE Computer Society, 2004.

[35] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B. Gibbons, and Christos Faloutsos. LOCI: fast outlier detection using the local correlation integral. In Umeshwar Dayal, Krithi Ramamritham, and T. M. Vijayaraman, editors, *Proceedings of the 19th International Conference on Data Engineering, March 5-8, 2003, Bangalore, India*, pages 315–326. IEEE Computer Society, 2003.

[36] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[37] Bernhard Schölkopf, Robert C Williamson, Alexander J Smola, John Shawe-Taylor, John C Platt, et al. Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588. Citeseer, 1999.

[38] Eleazar Eskin. Anomaly detection over noisy data using learned probability distributions. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 255–262. Morgan Kaufmann, 2000.

[39] Stephen Roberts and Lionel Tarassenko. A probabilistic resource allocating network for novelty detection. *Neural Computation*, 6(2):270–284, 1994.

[40] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Gregoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Muller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, May 2021.

[41] Henrique O Marques, Ricardo JGB Campello, Jörg Sander, and Arthur Zimek. Internal evaluation of unsupervised outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(4):1–42, 2020.

[42] Xiaoyi Gu, Leman Akoglu, and Alessandro Rinaldo. Statistical analysis of nearest neighbor methods for anomaly detection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

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

[44] G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, I. Assent E. Schubert, 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.

[45] Andrew Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. A meta-analysis of the anomaly detection problem.

[46] Erich Schubert, Arthur Zimek, and Hans-Peter 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.

[47] Andrew F Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. Systematic construction of anomaly detection benchmarks from real data. In *Proceedings of the ACM SIGKDD workshop on outlier detection and description*, pages 16–21, 2013.