Clustered Hierarchical Anomaly and Outlier Detection Algorithms

Thomas J. Howard III,[1*] **Najib Ishaq,** [1*] **Noah M. Daniels** [1]

[1] Dept. of Computer Science and Statistics
University of Rhode Island
Kingston, RI
thoward27@uri.edu, najib_ishaq@uri.edu, noah_daniels@uri.edu

Abstract

Abstract is written last.

Introduction

TODO

Related Works

TODO

Clustering Based Local Outlier Factor (CBLOF) The CBLOF operator calculates the outlier score based on cluster-based local outlier factor.

CBLOF takes as an input the data set and the cluster model that was generated by a clustering algorithm. It classifies the clusters into small clusters and large clusters using the parameters alpha and beta. The anomaly score is then calculated based on the size of the cluster the point belongs to as well as the distance to the nearest large cluster.

Use weighting for outlier factor based on the sizes of the clusters as proposed in the original publication. Since this might lead to unexpected behavior (outliers close to small clusters are not found), it is disabled by default.Outliers scores are solely computed based on their distance to the closest large cluster center.

By default, kMeans is used for clustering algorithm instead of Squeezer algorithm mentioned in the original paper for multiple reasons. (He, Xu, and Deng 2003)

Connectivity-Based Outlier Factor (COF) Connectivity-Based Outlier Factor (COF) COF uses the ratio of average chaining distance of data point and the average of average chaining distance of k nearest neighbor of the data point, as the outlier score for observations. (Tang et al. 2002)

Histogram-Based Outlier Detection (HBOS) Histogram-Based outlier detection (HBOS) is an efficient unsupervised method. It assumes the feature independence and calculates the degree of outlyingness by building histograms. (Goldstein and Dengel 2012)

*These authors contributed equally to this work.

*These authors contributed equally to this work.

Isolation-Forest Outlier Detector (IFOREST) The IsolationForest 'isolates' observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature. (Liu, Ting, and Zhou 2012) (Liu, Ting, and Zhou 2008)

k-Nearest Neighbors (kNN) For an observation, its distance to its kth nearest neighbor could be viewed as the outlying score. It could be viewed as a way to measure the density. (Ramaswamy, Rastogi, and Shim 2000) (Angiulli and Pizzuti 2002)

Linear Model Deviation-base outlier Detection(LMDD) LMDD employs the concept of the smoothing factor which indicates how much the dissimilarity can be reduced by removing a subset of elements from the data-set. (Arning, Agrawal, and Raghavan 1996)

LOcal Correlation Integral (LOCI) LOCI is highly effective for detecting outliers and groups of outliers (a.k.a.micro-clusters), which offers the following advantages and novelties: (a) It provides an automatic, data-dictated cut-off to determine whether a point is an outlier—in contrast, previous methods force users to pick cut-offs, without any hints as to what cut-off value is best for a given dataset. (b) It can provide a LOCI plot for each point; this plot summarizes a wealth of information about the data in the vicinity of the point, determining clusters, micro-clusters, their diameters and their inter-cluster distances. None of the existing outlier-detection methods can match this feature, because they output only a single number for each point: its outlierness score.(c) It can be computed as quickly as the best previous methods (Papadimitriou et al. 2003)

Lightweight Online Detector of Anomalies (LODA) (Pevný 2016)

Local Outlier Factor (LOF) The anomaly score of each sample is called Local Outlier Factor. It measures the local deviation of density of a given sample with respect to its neighbors. It is local in that the anomaly score depends on how isolated the object is with respect to the surrounding neighborhood. More precisely, locality is given by k-nearest neighbors, whose distance is used to estimate the local density. By comparing the local density of a sample to the local densities of its neighbors, one can identify samples that have a substantially lower density than their neighbors. These are considered outliers. (Breunig et al. 2000)

Minimum Covariance Determinant (MCD) Detecting outliers in a Gaussian distributed dataset using Minimum Covariance Determinant (MCD): robust estimator of covariance.

The Minimum Covariance Determinant covariance estimator is to be applied on Gaussian-distributed data, but could still be relevant on data drawn from a unimodal, symmetric distribution. It is not meant to be used with multi-modal data (the algorithm used to fit a MinCovDet object is likely to fail in such a case). One should consider projection pursuit methods to deal with multi-modal datasets.

First fit a minimum covariance determinant model and then compute the Mahalanobis distance as the outlier degree of the data (Rousseeuw and Driessen 1999) (Hardin and Rocke 2004)

One-class Support Vector Machine (OCSVM) Estimate the support of a high-dimensional distribution. (Schölkopf et al. 2001)

Subspace Outlier Detection (SOD) Subspace outlier detection (SOD) schema aims to detect outlier in varying subspaces of a high dimensional feature space. For each data object, SOD explores the axis-parallel subspace spanned by the data object's neighbors and determines how much the object deviates from the neighbors in this subspace. (Kriegel et al. 2009)

Stochastic Outlier Selection (SOS) SOS employs the concept of affinity to quantify the relationship from one data point to another data point. Affinity is proportional to the similarity between two data points. So, a data point has little affinity with a dissimilar data point. A data point is selected as an outlier when all the other data points have insufficient affinity with it. (Janssens 2012)

CHAODA

TODO

Methods

CLAM

We present a Manifold-Mapping algorithm called CLAM (Clustered Learning of Approximate Manifolds). This is an extension of earlier work presented in (Ishaq, Student, and Daniels 2019).

To start, we need a dataset and a distance function on the points in that dataset. A Dataset is a collection of n-points in a D-dimensional embedding space.

$$\mathbf{X} = \{x_1 \dots x_n\}, x_i \in \mathbb{R}^D$$

A Distance Function takes two points in the dataset and deterministically produces a non-negative real number.

$$f : (\mathbb{R}^D, \mathbb{R}^D) \mapsto \mathbb{R}^+$$

We require any distance function to have the following properties:

$$\forall x \in X, f(x,x) = 0$$
$$\forall x, y \in X, f(x,y) = f(y,x)$$

The distance function may or may not obey the triangle-inequality.

Clustering: We start by building a divisive-hierarchical clustering of the data. This gives us a tree of Clusters, with the root containing every point in the dataset, and each leaf containing a single point from the dataset. The procedure is detailed in (Ishaq, Student, and Daniels 2019).

Some important Cluster properties to consider are:

- *Cardinality*, i.e. the number of points in a cluster.
- *Center*, i.e. the geometric median of points contained in a cluster.
- *Radius*, i.e. the distance to the farthest point from a center.
- *Local fractal dimension*, as described in (Ishaq, Student, and Daniels 2019).
- *Parent-Child ratios* of cardinality, radius, and local fractal dimension.
- *Exponential moving averages* of the parent-child ratios along a branch of the tree.

In particular, we use the parent-child ratios and the exponential moving averages of those ratios to help generalize our anomaly detection method from a small set of datasets to a large, distinct set of datasets.

Graphs: Clusters that are near each other in the embedding space sometimes have overlapping volumes, i.e. the distance between their centers is less than or equal to the sum of their radii. We define a Graph with the clusters as the nodes and with edges between overlapping clusters. For our purposes, a Graph also has the following additional properties:

- The clusters in the graph collectively contain every point in the dataset.
- Each point in the dataset is in exactly one cluster in the graph.

A Graph can be built from clusters at a fixed depth in the cluster-tree, or can contain clusters from multiple different depths in the tree. We say that the cardinality of a graph is the number of clusters in that graph.

The Manifold: According to the Manifold Hypothesis (Fefferman, Mitter, and Narayanan 2016), datasets that come from constrained generating processes and are embedded in a high-dimensional space actually only occupy a low-dimensional manifold in that embedding space.

The graphs discussed so far map this low dimensional manifold in the original embedding space. Different graph do this at different levels of local and/or global resolution. Our aim is to properly build such a graph, i.e. we want to be "properly zoomed-in" to the various regions of the manifold formed by the data. We can then apply various anomaly detection algorithms to these graphs. These algorithms will often also incorporate information from the tree.

We describe some algorithms in **??**. While these algorithms are themselves fairly simple, the real challenge is in selecting the right clusters for the graphs that the algorithms operate on. We will demonstrate CLAM to be such a powerful technique in Manifold-Mapping that even such simple algorithms as described in **??** more often than not outperform state-ot-the-art anomaly detection algorithms.

Individual Algorithms

Here we describe several simple methods for anomaly detection. Each of these methods uses a graph of clusters from CLAM to calculate an anomalousness score for each point in the dataset.

Relative Cluster Cardinality: We measure the anomalousness of a point by the cardinality of the cluster that point belongs to relative to the cardinalities of the other clusters in the graph. Points in the same cluster are considered equally anomalous and points in clusters with relatively low cardinalities are considered more anomalous than points in clusters with relatively high cardinalities.

Child-Parent Cardinality Ratio: As described in CHESS (Ishaq, Student, and Daniels 2019), a cluster is partitioned by using its two maximally distant points as poles. The points are split among children by whichever pole they are closer to. Consider the fraction of points in a cluster that are assigned to each child. If a child cluster only contains a small fraction of the points that its parent did, then we consider the points in that child cluster to be anomalous. These child-parent cardinality ratios are accumulated for each point down its branch in the tree, terminating when the child cluster is a node in the graph. Points with a low value of these accumulated ratios are considered more anomalous than points with a high value of these accumulated ratios.

Graph Neighborhood Size: Given the graph with clusters and edges, consider the number of clusters reachable from a starting cluster within a given graph distance k. We call this number the *graph-neighborhood* of the starting cluster. When k is relatively small compared to the diameter of the graph, we can consider the relative *graph-neighborhoods* of every cluster in the graph. Points in clusters with small graph-neighborhoods are considered more anomalous than points in clusters with large graph-neighborhoods.

Relative Subgraph Cardinality We define disconnected components of a graph by the property that no two clusters from different disconnected components have an edge between them. Consider the relative cardinalities of each component in much the same we we considered the relative cardinalities of clusters in the Cluster Cardinality method. Points in clusters in the same component are considered equally anomalous and points in clusters in relatively small components are considered more anomalous than points in clusters in relatively large components.

Meta Machine Learning

The heart of the problem with our methods of anomaly detection is building the right graph to represent the underlying manifold. One could try using every possible combination of clusters to form a graph but this quickly leads to combinatorial explosion. Instead, we must intelligently select those clusters that, when used to build the graph, perform best for anomaly detection.

AUC ROC is often used to measure the performance of anomaly detectors; we will choose clusters so as to maximize this measure. Our aim is to learn a function that takes a cluster and predicts contribution to AUC from that cluster. Any function of the following form will suffice.

$$f : Cluster \mapsto \mathbb{R}^+$$

We chose a simple Linear Regression to fill this role. Such a model needs some data to train with. To generate this data, we took a random sample of some of the datasets described in **??**. We generate CLAM Manifolds for these training datasets, use the Linear Regression model to learn from these datasets, and apply the results to an entirely different set of datasets.

We generate the initial training data by considering graphs built from clusters at a uniform depth in the tree. For each such graph, we calculate the means of the cardinality ratio, radius ratio, and local fractal dimension ratio, and the exponential moving averages of these ratios. These ratios are described in **??**. These ratios form the feature vector for one training sample. We apply a method described in **??** to the graph and obtain an AUC ROC. We then train the Linear Regression model to predict this AUC from the features extracted from the graph. We train a separate Linear Regression model for each method described in **??**.

Ensemble

Given the learned meta-ml model from the training datasets, we can use it to build graphs for any other dataset. In our case with linear regression as that meta-ml model, we use the cluster ratios and the associated regression constants to rank every cluster in a CLAM tree. These rankings are normalized by the cardinality of each cluster. The highest ranked clusters are then used to build a graph. This graph is then used with the corresponding individual algorithm to calculate anomaly scores for all points in the dataset. The scores from each individual algorithm are then combined into an ensemble. We present the AUC scores from this ensemble in the Results section.

Datasets

We sourced 24 datasets, all from Outlier Detection Datasets (ODDS) (Rayana 2016), for training CHAODA and testing its performance. All of these datasets are adapted from the UCI Machine Learning Repository (UCIMLR) (Dua and Graff 2017), and have been standardized, by ODDS, for anomaly and outlier detection benchmarks.

The **annthyroid** dataset is derived from the "Thyroid Disease" dataset from the UCIMLR. The original data has 7200 instances with 15 categorical attributes and 6 real-valued attributes. The class labels are "normal", "hypothyroid", and "subnormal". For anomaly detection, the "hypothyroid" and "subnormal" classes are combined into 534 outlier instances, and only the 6 real-valued attributes are used.

The **arrhythmia** dataset is derived from “Arrhythmia“ dataset from the UCIMLR. The original dataset contains 452 instances with 279 attributes. There are five categorical attributes which are discarded, leaving this as a 274-dimensional dataset. The instances are divided into 16 classes. The eight smallest classes collectively contain 66 instances and are combined into the outlier class.

The **breastw** dataset is also derived from the “Breast Cancer Wisconsin (Original)“ dataset. This is a 9-dimensional dataset containing 683 instances of which 239 represent malignant tumors and are treated as the outlier class.

The **cardio** dataset is derived from the “Cardiotocography” dataset. The dataset is composed of measurements of fetal heart rate and uterine contraction features on cardiotocograms. The are each labelled “normal”, “suspect”, and “pathologic” by expert obstetricians. For anomaly detection, the “normal” class forms the inliers, the “suspect” class is discarded, and the “pathologic” class is downsampled to 176 instances forming the outliers. This leaves us with 1831 instances with 21 attributes in the dataset.

The **cover** dataset is derived from the “Covertype” dataset. The original dataset contains 581,012 instances with 54 attributes. The dataset is used to predict the type of forest cover solely from cartographic variables. The instances are labelled into seven different classes. For outlier detection, we use only the 10 quantitative attributes, type 2 (lodgepole pine) as the inliers, and type 4 (conttonwood/willow) as the outliers. The remaining classes are discarded. This leaves us with a 10-dimensional dataset with 286,048 instances of which 2,747 are outliers.

The **glass** dataset is derived from the “Glass Identification” dataset. The study of classification of types of glass was motivated by criminological investigations where glass fragments left at crime scenes were used as evidence. This dataset contains 214 instances with nine attributes. While there are several different types of glass in this dataset, class 6 is a clear minority with only nine instances and, as such, points in class 6 are treated as the outliers while all other classes are treated as inliers.

The **http** dataset is derived from the original KDD Cup 1999 dataset. It contains 41 attributes (34 continuous and 7 categorical) which are reduced to 4 attributes (service, duration, src_bytes, dst_bytes). Only the “service” attribute is categorical, dividing the data into http, smtp, ftp, ftp_data, others subsets. Here, only the “http” data is used. The values of the continuous attributes are centered around 0, so they have been log-transformed far away from 0. The original data contains 3,925,651 attacks in 4,898,431 records. This smaller dataset is created with only 2,211 attacks in 567,479 records.

The **ionosphere** dataset consists 351 instances with 34 attributes. One of the attributes is always 0 and, so, is discarded, leaving us with a 33-dimensional dataset. The data comes from radar measurements of the ionosphere from a system located in Goose Bay, Labrador. The data are classified into “good” if the radar returns evidence some type of structure in the ionosphere, and “bad” if not. The “good” class serves as the inliers and the “bad” class serves as the outliers.

The **lympho** dataset is derived from the Lymphography dataset. The data contain 148 instances with 18 attributes. The instances are labelled “normal find”, “metastases”, “malign lymph”, and “fibrosis”. The two minority classes only contain a total of six instances, and are combined to form the outliers. The remaining 142 instances form the inliers.

The **mammography** dataset is derived from the original Mammography dataset provided by Aleksandar Lazarevic. Its goal is to use x-ray images of human breasts to find calcified tissue as an early sign of breast cancer. As such, the “calcification” class is considered as the outlier class while the “non-clacification” class is the inliers. We have 11,183 instances with 6 attributes, of which 260 are “calcifications.”

The **mnist** dataset is derived from the classic MNIST dataset of handwritten digits. Digit-zero is considered the inlier class while 700 images of digit-six are the outliers. Furthermore, 100 pixels are randomly selected as features from the original 784 pixels.

The **musk** dataset is derived from its namesake in the UCI Machine Learning Repository. It is created from molecules that have been classified by experts as “musk” or “non-musk”. The data are downsampled to 3,062 instances with 166 attributes. The “musk” class forms the outliers while the “non-musk” class forms the inliers.

The **optdigits** dataset is derived from the Optical Recognition of Handwritten Digits dataset. Digits 1–9 form the inliers while 150 samples of digit-zero form the outliers. This gives us a dataset of 5,216 instances with 64 attributes.

Results

TODO

Discussion

TODO

Acknowledgements

TODO

References

Angiulli, F.; and Pizzuti, C. 2002. Fast Outlier Detection in High Dimensional Spaces. In Elomaa, T.; Mannila, H.; and Toivonen, H., eds., *Principles of Data Mining and Knowledge Discovery*, 15–27. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-45681-0.

Arning, A.; Agrawal, R.; and Raghavan, P. 1996. A Linear Method for Deviation Detection in Large Databases. 164–169.

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. LOF: Identifying Density-Based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD ’00, 93–104. New York, NY, USA: Association for Computing Machinery. ISBN 1581132174. doi:10.1145/342009.335388. URL https://doi.org/10.1145/342009.335388.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml.

Fefferman, C.; Mitter, S.; and Narayanan, H. 2016. Testing the manifold hypothesis. *Journal of the American Mathematical Society* 29(4): 983–1049.

Table 1: Performance on Train Datasets

dataset	annthyroid	mnist	pendigits	satellite	shuttle	thyroid
ensemble-L2	0.62	0.63	0.77	**0.63**	0.84	0.87
ensemble-L1	0.62	0.59	**0.96**	0.41	0.81	**0.96**
CBLOF	**0.76**	0.56	0.64	0.58	**0.98**	**0.96**
COF	0.51	0.58	0.55	0.54	*time*	0.52
HBOS	0.62	0.51	0.83	**0.65**	0.96	0.88
IFOREST	0.65	0.60	0.92	**0.64**	**0.98**	0.94
KNN	0.72	0.62	0.56	0.57	0.60	0.93
LMDD	0.70	*time*	0.69	0.44	*time*	0.93
LOCI	*time*	*time*	*time*	*time*	*time*	*time*
LODA	0.53	0.62	0.77	**0.64**	0.53	0.61
LOF	0.52	0.55	0.55	0.53	0.53	0.54
MCD	0.73	**0.67**	0.60	**0.65**	**0.96**	0.94
OCSVM	0.71	0.63	0.80	**0.64**	**0.98**	0.94
SOD	0.69	0.57	0.58	0.56	*time*	0.87
SOS	0.50	0.51	0.51	0.48	*time*	0.50

Table 2: Performance on the first half of the Test Datasets

dataset	arrhythmia	breastw	cardio	cover	glass	http	ionosphere	lympho	mammography
ensemble-L2	**0.72**	**0.93**	0.66	0.82	**0.82**	**1.00**	**0.81**	0.92	0.62
ensemble-L1	0.63	0.85	0.50	0.75	0.76	**0.99**	0.79	0.87	**0.81**
CBLOF	**0.71**	0.63	0.70	0.72	0.50	0.96	0.64	0.88	0.65
COF	0.64	0.43	0.54	*time*	0.56	*time*	0.63	0.71	0.60
HBOS	0.68	0.64	0.73	0.62	0.50	0.97	0.45	**0.97**	0.68
IFOREST	0.65	0.64	0.67	0.83	0.56	0.96	0.64	**0.97**	0.73
KNN	0.69	0.59	0.58	0.52	0.56	0.48	0.63	**0.98**	0.68
LMDD	0.66	0.64	0.70	*time*	0.45	*time*	0.62	0.71	0.77
LOCI	**0.70**	*time*	*time*	*time*	0.52	*time*	0.64	0.82	*time*
LODA	0.64	0.64	0.71	**0.88**	0.56	0.48	0.61	0.53	0.65
LOF	0.67	0.44	0.52	0.55	0.57	0.47	0.59	**0.97**	0.63
MCD	0.70	0.64	0.71	0.52	0.45	0.96	0.64	0.80	0.49
OCSVM	**0.71**	0.64	**0.75**	*time*	0.50	0.96	0.64	**0.97**	0.75
SOD	0.62	0.62	0.57	*time*	0.56	*time*	0.64	0.62	0.63
SOS	0.51	0.50	0.50	*time*	0.50	*time*	0.64	0.62	*time*

Goldstein, M.; and Dengel, A. 2012. Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm.

Hardin, J.; and Rocke, D. 2004. Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. *Computational Statistics and Data Analysis* 44(4): 625–638. ISSN 0167-9473. doi:10.1016/S0167-9473(02)00280-3.

He, Z.; Xu, X.; and Deng, S. 2003. Discovering cluster-based local outliers. *Pattern Recognition Letters* 24(9): 1641 – 1650. ISSN 0167-8655. doi:https://doi.org/10.1016/S0167-8655(03)00003-5. URL http://www.sciencedirect.com/science/article/pii/S0167865503000035.

Ishaq, N.; Student, G.; and Daniels, N. M. 2019. Clustered Hierarchical Entropy-Scaling Search of Astronomical and Biological Data. In *2019 IEEE International Conference on Big Data (Big Data)*, 780–789. IEEE.

Janssens, J. 2012. *Outlier Selection and One-Class Classification*. Ph.D. thesis.

Kriegel, H.-P.; Kröger, P.; Schubert, E.; and Zimek, A. 2009.

Table 3: Performance on the second half of the Test Datasets

dataset	**musk**	**optdigits**	**pima**	**satimage-2**	**smtp**	**vertebral**	**vowels**	**wbc**	**wine**
ensemble-L2	**1.00**	0.58	0.45	0.95	**0.86**	0.41	0.69	0.76	0.71
ensemble-L1	**1.00**	0.49	**0.60**	**0.99**	0.82	**0.46**	0.81	0.78	0.70
CBLOF	0.64	0.47	0.54	0.96	0.82	**0.46**	0.72	0.82	0.45
COF	0.53	0.52	0.54	0.61	*time*	**0.46**	0.89	0.78	0.45
HBOS	0.96	**0.75**	0.57	0.93	0.81	**0.46**	0.57	**0.85**	0.77
IFOREST	0.96	0.53	0.56	0.95	0.83	**0.46**	0.62	0.82	0.61
KNN	0.59	0.48	0.54	0.80	0.82	0.45	**0.94**	0.80	0.45
LMDD	0.96	0.45	0.53	0.45	*time*	0.44	0.51	0.75	0.66
LOCI	*time*	*time*	*time*	*time*	*time*	**0.47**	*time*	0.78	0.53
LODA	0.96	0.46	0.54	0.94	0.82	0.44	0.58	0.82	0.66
LOF	0.65	0.54	0.51	0.60	0.59	**0.46**	0.80	0.83	0.61
MCD	0.96	0.45	0.54	0.96	0.83	0.44	0.51	0.80	**0.88**
OCSVM	0.96	0.45	0.54	0.93	*time*	0.44	0.59	0.82	0.50
SOD	0.62	0.48	0.55	0.69	*time*	**0.46**	0.82	0.80	0.45
SOS	0.49	0.52	0.52	0.50	*time*	**0.48**	0.61	0.52	0.45

Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data. In Theeramunkong, T.; Kijsirikul, B.; Cercone, N.; and Ho, T.-B., eds., *Advances in Knowledge Discovery and Data Mining*, 831–838. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-01307-2.

Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation Forest. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, 413–422. USA: IEEE Computer Society. ISBN 9780769535029. doi:10.1109/ICDM.2008.17. URL https://doi.org/10.1109/ICDM.2008.17.

Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2012. Isolation-Based Anomaly Detection. *ACM Trans. Knowl. Discov. Data* 6(1). ISSN 1556-4681. doi:10.1145/2133360.2133363. URL https://doi.org/10.1145/2133360.2133363.

Papadimitriou, S.; Kitagawa, H.; Gibbons, P. B.; and Faloutsos, C. 2003. LOCI: fast outlier detection using the local correlation integral. *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405)* 315–326.

Pevný, T. 2016. Loda: Lightweight on-line detector of anomalies. *Machine Learning* 102(2): 275–304. ISSN 1573-0565. doi:10.1007/s10994-015-5521-0. URL https://doi.org/10.1007/s10994-015-5521-0.

Ramaswamy, S.; Rastogi, R.; and Shim, K. 2000. Efficient Algorithms for Mining Outliers from Large Data Sets. volume 29, 427–438. doi:10.1145/335191.335437.

Rayana, S. 2016. ODDS Library. URL http://odds.cs.stonybrook.edu.

Rousseeuw, P.; and Driessen, K. 1999. A Fast Algorithm for the Minimum Covariance Determinant Estimator. *Technometrics* 41: 212–223. doi:10.1080/00401706.1999.10485670.

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

Tang, J.; Chen, Z.; Fu, A. W.-C.; and Cheung, D. W.-L. 2002. Enhancing Effectiveness of Outlier Detections for Low Density Patterns. In *Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '02, 535–548. Berlin, Heidelberg: Springer-Verlag. ISBN 3540437045.