Pattern Analysis and Applications (2024) 27:61
https://doi.org/10.1007/s10044-023-01208-1

ORIGINAL ARTICLE

Understanding the limitations of self-supervised learning for tabular anomaly detection

Kimberly T. Mai[1,2] · **Toby Davies**[2,3] · **Lewis D. Griffin**[1]

Received: 15 September 2023 / Accepted: 28 December 2023 / Published online: 12 March 2024

Abstract
While self-supervised learning has improved anomaly detection in computer vision and natural language processing, it is unclear whether tabular data can benefit from it. This paper explores the limitations of self-supervision for tabular anomaly detection. We conduct several experiments spanning various pretext tasks on 26 benchmark datasets to understand why this is the case. Our results confirm representations derived from self-supervision do not improve tabular anomaly detection performance compared to using the raw representations of the data. We show this is due to neural networks introducing irrelevant features, which reduces the effectiveness of anomaly detectors. However, we demonstrate that using a subspace of the neural network's representation can recover performance.

Keywords Anomaly detection · Deep learning · Self-supervised learning · Tabular data

1 Introduction

Anomaly detection is the task of identifying unusual instances. Two issues hinder performance: how to obtain a "good" representation of the normal data and a lack of knowledge about the nature of anomalies. The emergence of self-supervised learning techniques has primarily addressed these issues in complex domains such as computer vision and natural language processing [1, 2]. However, these techniques have not yielded the same benefits for tabular data [3].

Self-supervised learning typically uses a pretext task to learn the intrinsic structure of the training data [4]. Examples of pretext tasks include colourising greyscale images [5] or predicting the next word in a sentence [6, 7]. Understanding the typical characteristics of a domain allows one to choose an effective pretext task. For instance, colourisation requires knowledge of object boundaries and semantics. These aspects are useful for image classification [8, 9]. However, unlike images or text where spatial or sequential biases are natural starting points for self-supervision, the starting points for tabular data are unclear.

A recent study indicated that self-supervised learning does not help tabular anomaly detection [3]. Reiss et al. compared two self-supervised methods with k-nearest neighbours (k-NN) on the original features. Even though the methods were designed for tabular data, they found that k-NN on the original features worked the best.

We seek to understand *why* this is the case. We extend the experiments to include a more comprehensive suite of pretext tasks. We also incorporate synthetic test cases and analyse the underlying learnt representations. Our results reinforce that self-supervision does not improve tabular anomaly detection performance and indicate deep neural networks introduce redundant features, which reduces the effectiveness of anomaly detectors. Conversely, we can recover performance using a subspace of the neural network's representation. We also show that self-supervised learning can outperform the original representation in the case of purely localised anomalies and those with different dependency structures.

✉ Kimberly T. Mai
kimberly.mai@ucl.ac.uk

Toby Davies
t.davies@leeds.ac.uk

Lewis D. Griffin
l.griffin@ucl.ac.uk

[1] Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK

[2] Department of Security and Crime Science, University College London, Gower Street, London WC1E 6BT, UK

[3] School of Law, University of Leeds, Woodhouse, Leeds LS2 9JT, UK

In addition to the above investigations, we ran a series of experiments to benchmark anomaly detection performance in a setting where we do not have access to anomalies during training. We include our findings as a complement to the self-supervision results and to provide practical insight into scenarios where specific detectors work better than others.

Our contributions are as follows:

1. We reconfirm the ineffectiveness of self-supervision for tabular anomaly detection.
2. We empirically investigate why self-supervision does not benefit tabular anomaly detection.
3. We introduce a comprehensive one-class anomaly detection benchmark using several self-supervised methods.
4. We provide practical insights and identify instances where particular anomaly detectors and pretext tasks may be beneficial.

In Sect. 2, we cover the anomaly detection setup. We proceed to outline our experimental approach in Sect. 3. We evaluate our findings in Sect. 4. Finally, we summarise our work and conclude in Sect. 5.

2 Background

2.1 Anomaly detection

Anomaly detection can be characterised as follows:

Let $\mathcal{X} \in \mathbb{R}^d$ represent the data space. We assume the normal data are drawn from a distribution $\mathcal{P}$ on $\mathcal{X}$. Anomalies are data points $\mathbf{x} \in \mathcal{X}$ that lie in a low probability region in $\mathcal{P}$. Therefore, the set of anomalies can be defined as follows [10]:

$$\mathcal{A} = \{\mathbf{x} \in \mathcal{X} | p(\mathbf{x}) \leq \tau\}, \quad \tau \geq 0, \tag{1}$$

Where τ is a threshold. Often, the original input space is not used as anomaly detection performance can be improved by using a different representation space. In the context of deep learning, a neural network parameterised by $\theta : \mathcal{X} \mapsto \mathcal{Y}$ (where $\mathcal{Y} \in \mathbb{R}^m$) is used to transform the input data. The anomalies are assumed to lie in a low-probability region in the new space. Namely, $\mathcal{P}$ on $\mathcal{X}$ transforms to $\mathcal{P}'$ on $\mathcal{Y}$ according to $\mathcal{P}'(\theta(\mathbf{x})) = |\mathbf{J}_\theta|$, where $\mathbf{J}$ is the Jacobian of θ. If θ is an effective mapping, then $\theta(\mathcal{A})$ will still be a low probability of $\mathcal{P}'$ and $\theta(\mathcal{A})$ will have a simpler boundary in $\mathcal{Y}$ than $\mathcal{A}$ in $\mathcal{X}$.

There are deep anomaly detectors (which aim to simultaneously transform the data to a new subspace and classify it) and shallow anomaly detectors (which do not transform the data but solely rely on an existing representation). This paper focuses on shallow anomaly detectors to isolate the differences in representations derived from different self-supervision tasks. Evaluating the transformative properties of deep anomaly detectors is out of scope. In addition, recent approaches suggest state-of-the-art anomaly detection performance is achievable by separating the representation learning and detection components [3, 11–14]. In this setup, we also assume only normal samples are present in the training set. This is referred to as a "one-class" setting in anomaly detection literature. The expressions "one-class learning" and "anomaly detection" are synonymous [10, 15, 16]. We use the same terms for consistency with the literature. We describe the anomaly detectors used in our analyses below. For a more detailed overview of anomaly detection techniques, we refer the reader to Ruff et al. [10].

k-**NN** assumes normal data closely surround other similar samples in the feature space, while anomalies have relatively fewer nearby neighbours. Despite being a simple approach, *k*-NN remains competitive in big data instances [3, 13, 14, 17, 18]. *k*-NN typically uses features extracted from pre-trained classification neural networks [13, 14, 18] for image-based anomaly detection. However, equivalent neural networks for tabular data do not exist.

Local outlier factor (LOF) is a density-based outlier detection method [19]. It compares the local density of a data point against its *k*-nearest neighbours. If the point's density is significantly lower, it is deemed anomalous.

Isolation forest (iForest) is an ensemble-based algorithm [20]. It uses a set of isolation trees. Each tree aims to isolate the training data into leaves. The tree construction algorithm randomly selects an attribute and a random split inside the attribute's range until each data point lies in a leaf. Each observation is assigned a score by calculating the length of the root node to the leaf and averaging across the trees. Points with shorter path lengths are considered more unusual, as the algorithm assumes anomalies are easier to isolate.

One-class support vector machine (OCSVM) assumes normal data lies in a high-density region [15]. Taking the origin as an anchor in the absence of anomalous data during training, it learns a maximum margin hyperplane that separates most training data from the origin. The algorithm considers a test datum's distance to the learnt hyperplane to classify anomalies. The method classifies a point as an anomaly if it lies on the side of the hyperplane closer to the origin.

Residual norms belong to the category of dictionary-based approaches. Dictionary-based approaches assume the building blocks of a feature space can reconstruct normal data but cannot construct anomalies. Methods using dictionaries use either linear or nonlinear manifold learning techniques (e.g. principal components analysis or autoencoders) to determine the building blocks [21–23]. We use the linear principal space approach from Wang et al. [23]

for our experiments. This technique achieves state-of-the-art results for out-of-distribution detection on images, verified in independent benchmarks [24]. Although introduced for images, the method itself is modality-neutral. We follow previous anomaly detection methodologies that have adapted image-based methods to other modalities while retaining acceptable performance [2, 3].

For consistency, we use Wang et al.'s [23] original notation and code implementation.[1] Given $\mathbf{X}$ as the in-distribution data matrix of training samples, we find the principal subspace $\mathbf{W}$ from the matrix $\mathbf{X}^T\mathbf{X}$. This subspace spans the eigenvectors of the D largest eigenvalues of $\mathbf{X}^T\mathbf{X}$. We assume anomalies have more variance on the components with smaller explained variance [25]. Therefore, we project $\mathbf{X}$ to the subspace spanned by the *smallest* eigenvalues of D (represented by $\mathbf{W}^{\perp}$) to encapsulate the residual space and take its norm as the anomaly score:

$$||\mathbf{x}^{W^{\perp}}||. \tag{2}$$

2.2 An overview of self-supervised learning

Self-supervision approaches devise tasks based on the intrinsic properties of the training data. By exploiting these properties, neural networks hopefully learn about the regularities of the data. Examples across different modalities include:

Classifying perturbations: Each training datum is subject to a perturbation randomly selected from a fixed set, such as rotating the input data [26] or reordering patches in an image [27]. A classification model then learns to predict which perturbation was applied.

Conditional prediction. A neural network sees pieces of the input data and learns to complete the remaining parts. Examples include predicting the next word given a portion of a sentence [6] or filling in masked areas of an image [28, 29].

Clustering. Under this category, models learn to group semantically similar instances and place them far away from observations representing other semantic categories. k-means clustering is a classic example that measures similarity in Euclidean space.

More modern techniques learn a similarity metric using neural mappings. One popular loss function that enables this is InfoNCE [30, 31]. InfoNCE takes augmented views of the same data point as positives and learns to group them while pushing away other data points. Variants of this method sample from the positive's nearest neighbours to create more semantic variations [32, 33]. Augmentations are usually in the form of transformations. In the case of images, these can involve adding noise, colour jittering, or horizontal flips. However, InfoNCE relies on large batch sizes to enable sufficiently challenging comparisons. Augmentation choices are also vital, as aggressive transformations could remove relevant semantic features.

VICReg [34] attempts to overcome some of the issues of InfoNCE by enforcing specific statistical properties. It encourages augmented views to have a high variance to ensure the neural mapping learns diverse aspects of the data. It also regularises the covariance matrix of the representations. This regularisation ensures the neural mapping covers complementary information across the representation space.

Additional pretext tasks are covered in more detail in Balestriero et al. [4].

2.3 Self-supervised learning and anomaly detection for non-tabular data

Anomaly detection for non-tabular data has benefited from self-supervision. Golan and El-Yaniv [35] show that compared to OCSVMs trained on pixel space, outputs from a convolutional neural network trained to predict image rotations were more reliable for anomaly detection. Mai et al. [2] demonstrate similar findings on text. They show that good anomaly detection performance is achievable by fine-tuning a transformer with a self-supervised objective and using the loss as an anomaly score.

Other successful approaches do not use a self-supervised model in an end-to-end manner for anomaly detection. The works of Sehwag et al. [12] and Tack et al. [11] both extract features from neural networks trained with an InfoNCE objective to perform anomaly detection on images. Sehwag et al. classify anomalies using the Mahalanobis distance on the extracted space, while Tack et al. use a product of cosine similarities and norms.

2.4 Self-supervised learning and anomaly detection for tabular data

Literature covering self-supervision for anomaly detection in tabular data is more limited. GOAD [36] extends the work of Golan and El-Yaniv [35] to a more generalised setting. They apply random affine transformations to the data and train a neural network to predict these transformations. At inference, they apply all possible transformations to the test data, obtain the prediction of each transformation from the network and aggregate the predictions to produce the anomaly score. The network should be able to predict the correct modification with higher confidence for the normal data versus the anomalies.

ICL [37] adapts the InfoNCE objective. It considers one sample at a time. Taking a sample $\mathbf{x}_i$ of dimensionality d, ICL splits $\mathbf{x}_i$ into two parts. The dimensionality of the two

[1] https://github.com/haoqiwang/vim.

Table 1 Summary of related self-supervised anomaly detection literature across modalities

Modality	Year	Author	Pretext task	Anomaly detector
Images	2018	Golan and El-Yaniv [35]	Rotation prediction	Classification confidence
	2020	Tack et al. [11]	Contrastive learning	Cosine similarity and norm
	2021	Sehwag et al. [12]	Contrastive learning	Mahalanobis distance
Text	2022	Mai et al. [2]	Masked language modelling Causal language modelling Contrastive learning	Loss
Tabular	2022	Shenkar and Wolf [37]	Contrastive learning	Loss
Multiple	2020	Bergman and Hoshen [36]	Transformation prediction	Classification confidence
	2022	Reiss et al. [3]	Contrastive learning Transformation prediction	k-NN

parts depends on a given window size, k ($k < d$). The first part $\mathbf{a}_i$ is a continuous section of size k, while the second $\mathbf{b}_i$ is its complement of size $d - k$. A Siamese neural network containing two heads with dimensionalities k and $d - k$ aims to push the representations together. The negatives are other contiguous segments of $\mathbf{x}_i$ of size k. As the neural network should be capable of aligning the normal data and not anomalies, the loss is the anomaly score.

Although both methods claim to be state-of-the-art for tabular anomaly detection, Reiss et al. [3] did not find this to be the case. They replicated the pipelines of GOAD and ICL. In addition, they used the trained neural networks of GOAD and ICL as feature extractors. After extracting the features, they ran k-NN on the new representations. They compared both setups to k-NN on the original data. Although GOAD and ICL are specifically designed to process tabular data, Reiss et al. found that k-NN on the original data was the best-performing approach. However, they did not run a hyperparameter search to optimise the choice of k (leaving it as $k = 5$). They also used the original architectures designed for GOAD and ICL, which differ from each other. This choice could be another confounding factor affecting results.

We summarise the works that cover self-supervision and anomaly detection in Table 1.

3 Method

3.1 Datasets

We use 26 multi-dimensional point datasets from Outlier Detection Datasets (ODDS) [38]. Each datum comprises one record, which contains multiple attributes. Table 2 summarises the properties of the datasets. We treat each dataset as distinct and train and test separate anomaly detection models for each dataset.

We follow the data split protocols described in previous tabular anomaly detection literature [36, 37]. We randomly select 50% of the normal data for training, with the remainder used for testing. The test split includes all anomalies. The training split did not use any anomalies as we adopt a one-class setup. We partition the training set further by leaving 20% for validation.

Table 2 Summary of ODDS datasets

Dataset	Total size	Number of anomalies (%)	Dimensionality
Annthyroid	7,200	534 (7.4%)	6
Arrhythmia	452	66 (14.6%)	274
BreastW	683	239 (35.0%)	9
Cardio	1,831	176 (9.6%)	9
Glass	214	9 (4.2%)	9
Heart	224	10 (4.4%)	44
HTTP	567,469	2,211 (0.4%)	3
Ionosphere	351	126 (35.8%)	33
Letter	1,600	100 (6.3%)	32
Lympho	148	6 (4.1%)	18
Mammography	11,183	260 (2.3%)	6
MNIST	7,603	700 (9.2%)	100
Musk	3,062	97 (3.2%)	166
Optdigits	5,216	150 (2.9%)	64
Pendigits	6,870	156 (2.3%)	16
Pima	768	268 (34.9%)	8
Satellite	6,435	2,036 (31.6%)	36
Satimage-2	5,803	71 (1.2%)	36
Seismic	2,584	170 (6.5%)	11
Shuttle	49,097	3,511 (6.6%)	9
SMTP	95,156	30 (0.03%)	3
Speech	3,686	61 (1.7%)	400
Thyroid	3,772	93 (2.4%)	6
Vertebral	240	30 (12.5%)	6
Vowels	1,456	50 (3.4%)	12
WBC	278	21 (5.6%)	30
Wine	129	10 (7.7%)	13

3.2 Baseline approach

We run k-NN, iForest, LOF, OCSVM, and residual norms on the original training data. As we aim to expand on the work of Reiss et al. [3], we only implement one-class detectors for comparability. Even though Reiss et al. [3] only use k-NN in their experiments, we use multiple detectors to establish whether k-NN is the best detector or if there are other more appropriate detectors depending on the type of anomalies present. We analyse our findings in Sect. 4.7. Another anomaly detection study, ADBench [39], follows a similar protocol. However, their setup assumes anomalies are present in the training data. Through our experiments, we establish whether a purely one-class setup affects overall detector ranking. We use scikit-learn [40] to implement all detectors except for k-NN, which uses the faiss library [41].

We also investigate the detectors' sensitivity to different configurations by varying the hyperparameters. For k-NN and LOF, we report results for $k = \{1, 2, 5, 10, 20, 50\}$. For the residual norms, we look at how results change with a proportion of features, with percentages ranging from 10% to 90% in 10% increments [10%, 20%, ..., 90%]. We record our findings in Sect. 4.7. For the self-supervised tasks, we report the results based on the best hyperparameter configuration derived from these ablations. We retain the default scikit-learn parameters for iForest and OCSVM, which uses a radial basis function kernel.

The detectors run directly on the data and on a standardised version. We standardise each dimension independently by removing the mean and scaling to unit variance. We also experimented with fully whitening the data but found attribute-wise standardisation rendered similar results.

3.3 Self-supervision

3.3.1 Pretext tasks

Although tabular data lack overt intrinsic properties like those in images or text, we choose self-supervised tasks that we hypothesise can take advantage of its structure.

Firstly, we adapt ICL [37] and GOAD [36] to use them as pretext tasks. We do not directly implement ICL and GOAD as they score anomalies in an end-to-end manner. In contrast, our experiments focus on how representations from different pretext tasks affect shallow detection performance. Therefore, we refer to the ICL-inspired task as "**EICL**" (embedding-ICL) for the remainder of the paper. As GOAD uses random affine transformations, we can consider this a combination of predicting rotation and stretches. This configuration conflates two different tasks and could be trivial to solve. Therefore, we attempt to align it closer to the RotNet [1, 26] experiments for image-based anomaly detection by training a model to classify orthonormal rotations. This pretext task should profit from the rotationally invariant property of tabular data [42]. Hence, we refer to the GOAD-inspired task as "**Rotation**".

The additional objectives used in the experiments are as follows:

Predefined shuffling prediction (Shuffle): We pick a permutation of the dimensions of the data from a fixed set of permutations and shuffle the order of the attributes based on the selection. The model learns to predict that permutation.

Predefined mask prediction (Mask classification): Given a mask rate r $(r < d)$, we initialise predefined classes that indicate which attributes to mask. We perform masking by randomly selecting another sample $\mathbf{x_j}$ from the training set and replacing the chosen attributes in $\mathbf{x_i}$ with those from $\mathbf{x_j}$. We follow the protocol outlined in Yoon et al. [43]. This approach generated better representations compared to alternative masking strategies like imputation, and constructing a mask classification pretext task outperformed alternative supervised and semi-supervised methods on tabular classification tasks. The model learns to classify which predefined class was applied.

Masked columns prediction (Mask columns): The model picks which attributes were masked given a mask rate r. For example, if only the first attribute was masked, a correct classification should identify the first attribute and should not pick the other attributes. This is different from the mask classification task, where the predefined mask class is given a label from a fixed set of combinations rather than from the particular attribute that has been masked. (For example, if there are only two classes, the labels for mask classification are 0 or 1.)

Denoising autoencoding (Autoencoder): Given a mask rate r, we perturb $\mathbf{x_i}$ by randomly selecting another sample $\mathbf{x_j}$ and replacing a subset of $\mathbf{x_i}$'s attributes with those of $\mathbf{x_j}$. The perturbed $\mathbf{x_i}$ is the input. Given this input, the model learns to reconstruct the unperturbed $\mathbf{x_i}$.

Contrastive learning: We create positive views of $\mathbf{x_i}$ by rotating the data using an orthonormal matrix **(Contrastive rotation)**, permuting the attributes per the shuffle task **(Contrastive shuffle)**, or masking the attributes per the mask classification task **(Contrastive mask)**. We treat other data points in a minibatch as negatives. We only apply one augmentation at a time to isolate their effects.

3.3.2 Network architectures and loss functions

We use the same neural network architectures to control for any potential effects on performance. Per the findings of Gorishniy et al. [44], we use ResNets [45] and FT-Transformers. Gorishniy et al. examined the performance of several deep learning architectures on tabular classification and regression, including multilayer perceptrons, recurrent neural networks, ResNets and transformers. Their results

Table 3 Summary of the model configurations

Anomaly detectors	Architectures	Self-supervised tasks	Loss functions
k-nearest neighbours	ResNet	Rotation	Cross-entropy
Isolation forest	FT-Transformer	Shuffle	ARPL
Local outlier factor		Mask classification	AAM
One-class support vector machine		Mask columns	Binary cross-entropy
Residual norms		Autoencoder	MSE
			MAE
		EICL	
		Contrastive - rotation	InfoNCE
		Contrastive - shuffle	VICReg
		Contrastive - mask	

indicated that ResNets and FT-Transformers were the best overall. Based on these findings, we restrict our architectures to the most promising variants. FT-Transformer is a transformer specially adapted for tabular inputs where each transformer layer operates on the feature level of one datum.

We train both architectures on all objectives except for EICL, where we only use ResNets. As EICL requires specific partitioning of the features, the FT-Transformer architecture would need to be modified. This modification is out of the scope of our experiments. We retain the same architecture (e.g. the number of blocks) for each pretext task and only vary the dimensionality of the output layer. The dimensionality corresponds to the number of preset classes for the rotation, shuffle, and mask classification tasks. The output dimensionality of the autoencoder task mirrors the input dimensionality. For the contrastive objectives (including EICL), we set the output as one of $\{128, 256, 512\}$ depending on validation performance.

As previous literature has claimed specialised loss functions can improve out-of-distribution detection on other modalities [46, 47], we examine these to confirm whether they also improve tabular anomaly detection.

For the rotation, shuffle, and mask classification tasks, we use cross-entropy, adversarial reciprocal points learning (ARPL) [46], and additive angular margin (AAM) [48]. ARPL is a specialised loss function for out-of-distribution detection. The probability of a datum belonging to a class is proportional to its distance to a reciprocal point. The point represents "otherness" in the learnt feature space. AAM is a loss function typically used for facial recognition. AAM specifically enforces interclass similarity and ensures interclass separation using a specified margin. This results in more spherical features for each class. We include AAM as some literature claims spherical per-class features make out-of-distribution detection easier [49]. Finally, we incorporate the cross-entropy loss as studies have shown models trained with this loss function can meet or outperform specialised losses like ARPL with careful hyperparameter selection [47]. We experiment with mean squared error and mean absolute error for the autoencoders. We use the binary cross-entropy loss for masked column prediction, as multiple masked columns correspond to more than one label for each datum. For the contrastive objectives, we experiment with both InfoNCE and VICReg.

We summarise all the possible model configurations in Table 3.

3.3.3 Model selection

Due to the number of potential hyperparameter combinations, we perform random searches to determine the most appropriate models for anomaly detection. We pick hyperparameters randomly and train on the training split for each self-supervised task and dataset. As we cannot evaluate using anomalies, we select models that achieve the lowest loss on the normal validation data. As we want to analyse the effect of different loss functions and architecture, the hyperparameter sweep stage results in a maximum of twelve configurations for each dataset and task. For example, the models trained on the rotation task would include ResNets and FT-Transformers, each architecture also includes the cross-entropy, ARPL, and AAM losses. There are also different configurations for standardised and non-standardised input data.

3.3.4 Feature extraction

After training, we obtain the learnt features by passing input data through the self-supervised models. We extract the features from the penultimate layer. As we fix the architecture for the different tasks, we obtain 128-dimensional embeddings for ResNets and 192-dimensional embeddings for FT-Transformer. We train the anomaly detectors using the new training features and test them using the transformed test

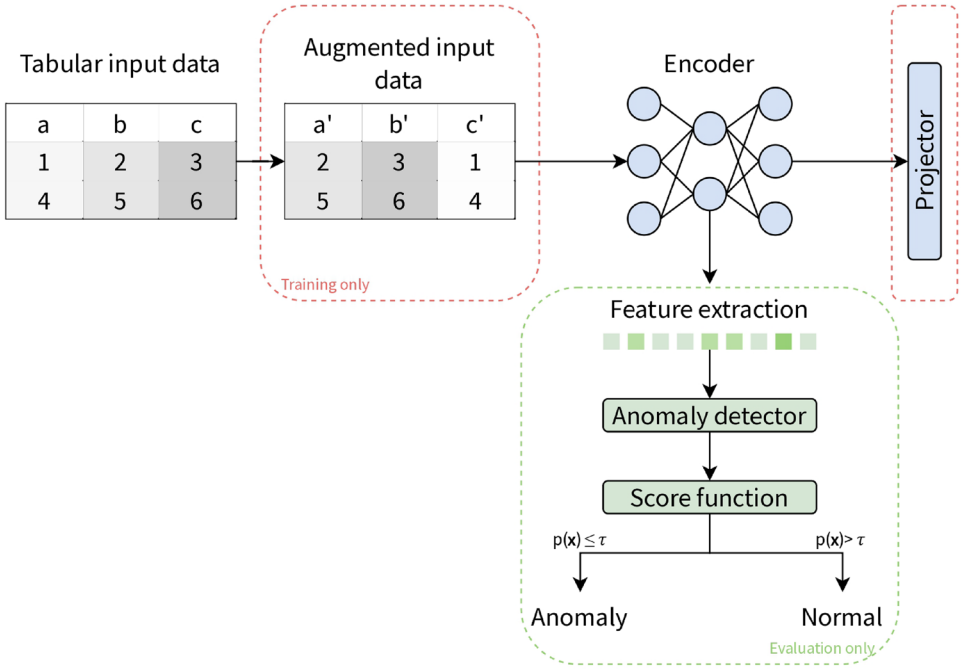


Fig. 1 Self-supervised anomaly detection workflow. The data are only augmented and fed through the projector during training

features. We do not apply any augmentations during inference to ensure a fair comparison between the self-supervised tasks. Figure 1 shows the workflow.

3.4 Evaluation

We evaluate all anomaly detectors using the area under the receiving operator curve (AUROC) score. We can consider AUROC as the probability that a randomly selected anomaly will be ranked as more abnormal than a normal sample. Scores fall between 0% and 100%. A score of 50% indicates the detector cannot distinguish between anomalies and normal data points, while a score of 100% signals perfect anomaly discrimination. We choose AUROC as it does not require a threshold to control for false positives, for example.

3.5 Additional ablations

In addition to evaluations with the ODDS dataset, we run more experiments to understand detector performance and scenarios where specific self-supervised objectives may perform better than others.

3.5.1 Synthesised anomalies

Although ODDS contains several datasets, the datasets may mix different types of anomalies. These mixes can make it difficult to diagnose why one representation performs better than another. Therefore, we evaluate how the pretext tasks and their learnt representations fare with synthesised anomalies. We keep the normal data in the train and test splits and only generate anomalies by perturbing the properties of the normal training data. We use the four synthetic anomaly categories as defined in ADBench [39, 50]. We use ADBench's code to create all types.

- **Local** anomalies deviate from their local cluster. We use Gaussian mixture models (GMM) to learn the underlying normal distribution. The covariance matrix undergoes scaling by a factor α to generate the anomalies. We use $\alpha = 2$ in our experiments.
- **Cluster** anomalies use GMMs to learn the normal distribution. A factor β scales the mean feature vector to create the cluster anomalies. We use $\beta = 2$ in our experiments.
- **Global** anomalies originate from a uniform distribution $U[\delta \cdot \min(\mathbf{X}_i^k), \delta \cdot \max(\mathbf{X}_i^k)]$. δ is a scaling factor, and the minimum and maximum values of an attribute $\mathbf{X}_i^k$ define the boundaries. We use $\delta = 0.01$.
- **Dependency** anomalies do not follow the regular dependency structure seen in normal data. We use vine copulas to learn the normal distribution and Gaussian kernel density estimators to generate anomalies.

3.5.2 Corrupted input data

Previous work hypothesises neural networks underperform on tabular classification and regression because of their rotational invariance and lack of robustness to uninformative features [42]. We investigate if this occurs for anomaly detection. Simultaneously, we explore the shallow anomaly detectors' sensitivity to corrupted attributes. Understanding these results can give a practical insight into what

self-supervision objectives and anomaly detectors work best when the data are noisy or incomplete. For our ablations, we follow Grinsztajn et al. [42] and apply the following corruptions to the raw data:

1. **Adding uninformative features**: We add extra attributes to **X**. We select a subset of attributes to imitate. We then generate features by sampling from a multivariate Gaussian based on the mean and interquartile range of the subset's values. We experiment with different proportions of additional features and limit the maximum number of extra attributes to be no greater than the existing number of features in the dataset.
2. **Missing values**: We randomly remove a proportion of the entries and replace the missing values using the mean of the attribute the value belongs to. We apply this transformation to both the train and test sets.
3. **Removing important features**: We train a random forest classifier to classify between normal samples and anomalies. We then drop a proportion of attributes based on the feature importance values output by the random forest, starting from the least important. This corruption violates the one-class assumption within our anomaly detection setup. However, we use this to analyse the robustness of the detectors and self-supervised models.
4. **Selecting a subset of features**: Similar to (3), we train a random forest classifier. We choose a proportion of attributes based on the feature importance values output from the random forest, starting from the most important.

After corrupting the data, we follow the same process of training the self-supervised models and feature extraction for the neural network experiments.

4 Results

We organise our results as follows: Sect. 4.1 reconfirms the ineffectiveness of self-supervision for tabular anomaly detection and summarises the main results at a high level. We investigate this phenomenon through a series of case studies and ablations. Sections 4.2 and 4.3 drill down on performance using a subset of ODDS (*HTTP*) and simplified toy scenarios. Our working hypothesis is that self-supervision introduces irrelevant directions. We empirically verify our hypothesis by investigating the residual space of the embeddings in Sect. 4.4. We attempt to compare the properties of the self-supervised pretext tasks by replacing ODDS anomalies with synthetic variants in Sect. 4.5. Finally, we investigate the effect of architecture and detector choices in Sect. 4.6.

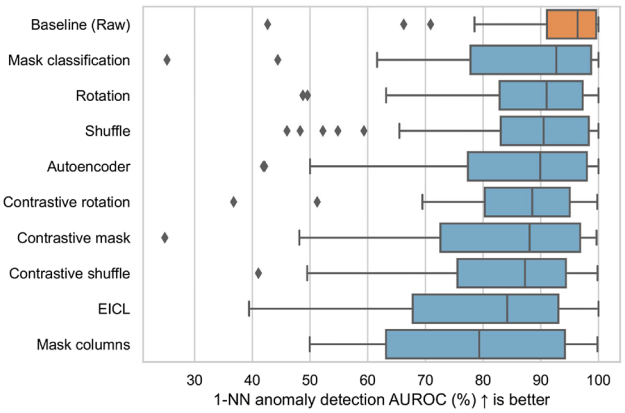


Fig. 2 Box plot comparing nearest neighbour AUROCs for each of the embeddings, ordered by median performance. For each self-supervised task, we filter the results by architecture and loss function to include the embedding with the best-performing results

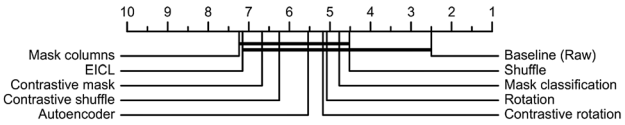


Fig. 3 Critical difference diagram comparing the embeddings in a pairwise manner. The horizontal scale denotes the average rank of each embedding. The dark lines between different detectors indicate a statistical difference ($p < 0.05$) in results when running pairwise comparison tests. The baseline scores greatly outrank the pretext tasks. In contrast, the scores among the pretext tasks are more closely aligned

4.1 Self-supervision results

No self-supervision task outperforms the baseline Figure 2 summarises the nearest neighbour performance derived from the embeddings of each self-supervised approach. We aggregate performance by representation rather than dataset to concentrate on the influence each representation has on performance. No self-supervision task exceeds k-NN on the raw tabular data. When comparing results at a pairwise level, Fig. 3 shows that the baseline scores greatly outrank the self-supervised objectives. Similarly, performance using the self-supervised embeddings drops in the presence of corrupted data (Appendix, Fig. 18). These results extend the findings in [42] that neural networks are also more sensitive to corrupted attributes in the anomaly detection task. When excluding the baseline, the classification-based tasks (shuffle, mask classification, and rotation) outperform their contrastive and reconstructive counterparts.

We observe similar results when we use different shallow detectors to perform anomaly detection (Fig. 4), with one exception. Using residual norms on the embedding space

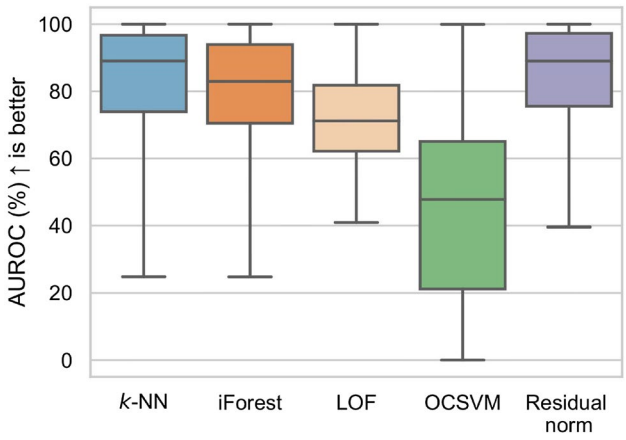


Fig. 4 Box plot comparing detector performance on the self-supervised embeddings

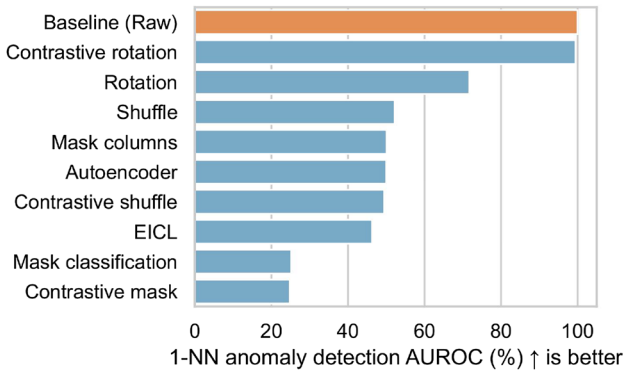


Fig. 5 Bar chart comparing baseline and self-supervised embedding results on HTTP

is a better choice than k-NN. However, they still lag behind k-NN scores on the original embeddings. We also observe that OCSVM performs consistently worse across all tasks.

4.2 A case study on HTTP

To understand why self-supervision does not help, we will explore one ODDS dataset in detail. We proceed to test our reasoning on toy datasets and then analyse the remaining ODDS datasets.

We use *HTTP* for our analyses. *HTTP* is a modified subset of the KDD Cup 1999 competition data [51]. The competition task involved building a detector to distinguish between intrusive (attack) and typical network connections. The dataset initially contained 41 attributes from different sources, including HTTP, SMTP, and FTP. The ODDS version only uses the "service" attribute from the *HTTP* information as it is considered one of the most basic features. The resulting subset is three-dimensional and comprises over 500,000 observations. Out of these samples, 2,211 (0.4%) are attacks.

It is easy to find attacks when running detectors directly on the raw ODDS variant of *HTTP*. In our experiments, all shallow methods achieve AUROCs between 87.9% and 100% on non-standardised data, with the median score being 99.7%. Further investigations show the attacks are separate from typical connections. A supervised logistic regression model trained to classify the two classes achieves 99.6% AUROC, even with only 200 sample anomalies for training.

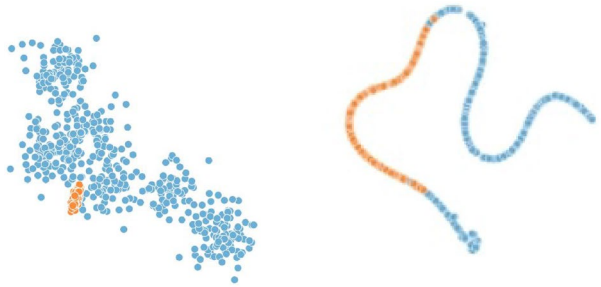
However, we observe peculiar results when using representations devised from the pretext tasks for *HTTP*. k-NN performance drops drastically across the majority of tasks (Fig. 5), sometimes yielding scores worse than random. Conversely, the other detectors maintain their performance. For example, when extracting features from the rotation task,[2] k-NN obtains 71.8% AUROC, while iForest, OCSVM, and residual norms preserve AUROCs around 99%. In addition, logistic regression continues to classify anomalies with 99% AUROC in the supervised setting using the rotation task representations. As k-NN is susceptible to the curse of dimensionality, these initial results suggest the neural network representation introduces directions that obscure informative distances between the typical and intrusive samples. Moreover, as iForest uses a splitting strategy for detection, its consistent results indicate *some* direction signalling anomalousness exists.

4.3 Toy data analysis

It can be challenging to draw conclusions based on existing datasets, as they are large and often contain uninterpretable features. Therefore, we pivot to toy examples to understand these behaviours. We devise nine two-dimensional toy datasets of varying difficulty (Appendix, Fig. 19). Like the experiments on the ODDS, we first evaluate performance directly on the two-dimensional representations. We then train ResNets on a two-class rotation prediction task, extract features from the penultimate embedding and re-run the detectors on the new space. We use this setting as rotations can be performed on two-dimensional data, and ResNets require less compute than the FT-Transformers. We apply the same architecture as the ODDS experiments, making the extracted features 128-dimensional.

Regardless of whether the network can or cannot identify the rotation applied to the data, we observe behaviours consistent with ODDS in most toy instances. Compared to the original two-dimensional results, detection performance drops for almost all detectors after extracting representations from the ResNets. As two dimensions are sufficient to capture the characteristics of the datasets, projecting the

[2] Using the best-performing rotation model, which is an FT-Transformer trained with ARPL loss.



(a) Original 2D data (b) t-SNE visualisation of the self-supervised features

Fig. 6 Visualisations of the multiple Gaussian toy dataset. Light blue are the normal data and orange are the anomalies. The features extracted from the neural network appear to be more narrow (**b**) and stretched compared to their original 2D representation (**a**)

data to a 128-dimensional space only results in a stretched and narrow representation without extra information. The t-SNE plots highlight this activity. We show an example of the multiple Gaussian dataset in Fig. 6.

We project the embeddings extracted from the ResNets to a lower dimensional space using the residual eigenvectors from the training data to verify whether the curse of dimensionality affects performance. We conduct this projection because the residual norm method outperforms *k*-NN in the self-supervised experiments. Therefore, we hypothesise that projecting to a smaller space should reduce the distracting influence of the primary principal components. Consequently, running shallow detectors in this new space should garner improvements. We discard half of the directions for the toy experiments to form 64-dimensional embeddings. The anomaly detectors perform better in this new space (Fig. 7), corroborating the view that the neural network embeddings introduce irrelevant directions.

We can also use the toy scenarios to attempt to understand the behaviour of the detectors such as OCSVM. Our experiments suggest OCSVM fails when anomalies lie in the centre of the normal data. For example, the AUROC for OCSVM trained on the raw ring data signalled random performance at 50%, whereas *k*-NN could detect the anomalies perfectly.

4.4 Analysing ODDS embeddings

We now proceed to run ablations on ODDS. Previous studies have shown that supervised classification performance correlates highly with out-of-distribution detection performance [47]. Therefore, we train linear classifiers on the self-supervised and original representation and compare classification performance. If there is a drop in performance on the self-supervised embeddings, the results would suggest the neural networks transform the data in a way that mixes anomalies

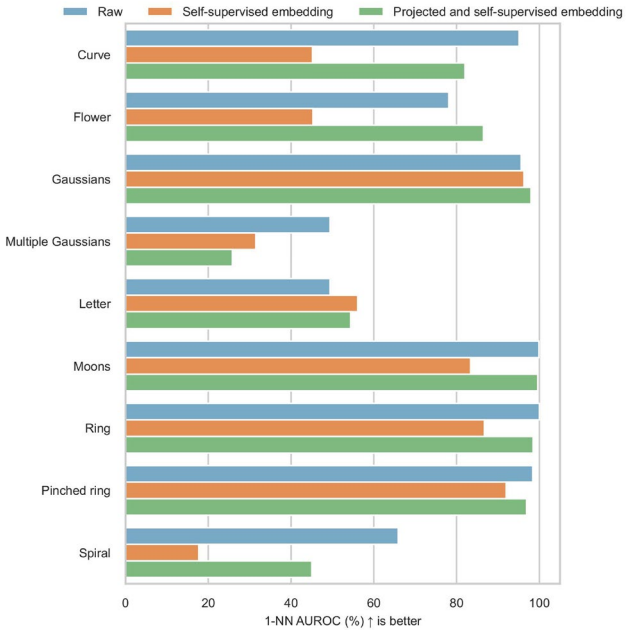


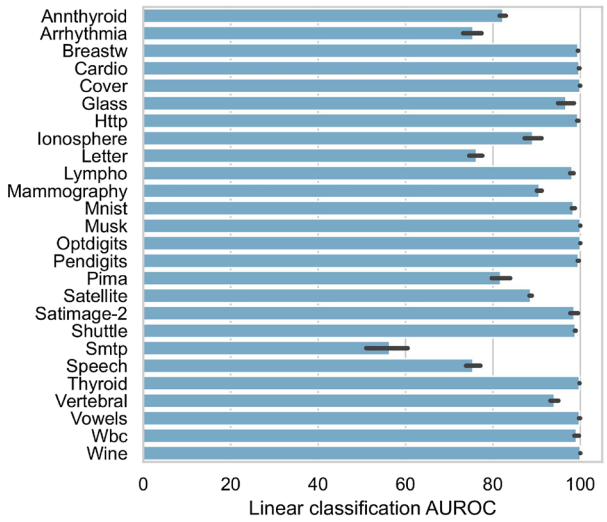
Fig. 7 Nearest neighbour performance on the toy datasets. The raw embedding (blue) is the best in almost all instances. However, the self-supervision embeddings (orange) improve when projecting to a lower dimensional space (green)

with the normal samples. We could consequently attribute the poor self-supervised performance to this mixing rather than the presence of irrelevant directions.

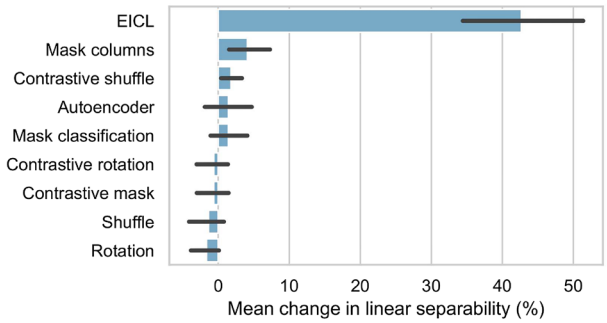
Figure 8a illustrates classification scores on the raw data. Most datasets are almost perfectly linearly separable in this embedding space, indicating that anomaly detectors should perform well. Figure 8b depicts the mean difference between the raw and self-supervised classification performances. Except for EICL, the differences between linear classification performance on the raw embeddings and the self-supervised embeddings are close to zero. These trends suggest the self-supervised embeddings retain reasonable separability between the normal data and anomalies. We can rule out the mixing effect and conclude that self-supervision generally does not affect the separability between the two classes.

We now investigate the residual space of the embeddings by extending the toy dataset analyses to ODDS. We take the smallest eigenvalues (from 1% to 90% in 10% increments) to project the neural network embeddings to their residual representations. We proceed to re-run the shallow anomaly detectors in the new space. Figure 9 shows the results. We aggregate both ResNet and FT-Transformer scores as observed similar behaviour across the two architectures. Reducing the dimensionality indeed boosts performance.

On all of the shallow detectors, using the entire representation space (100% dimensionality in Fig. 9) results in lower AUROCs than using a subset. Throwing away the top 10% of principal components garners most improvements, although



(a) Classification results on the raw embeddings.



(b) Differences between linear classification performance on the raw embeddings compared to the self-supervised embeddings, aggregated across the ODDS datasets. Changes greater than 0 mean the self-supervision embedding reduced separability.

Fig. 8 Supervised linear classification results (normal versus anomaly) on raw data (**a**) and supervised classification comparisons against the self-supervised embeddings (**b**)

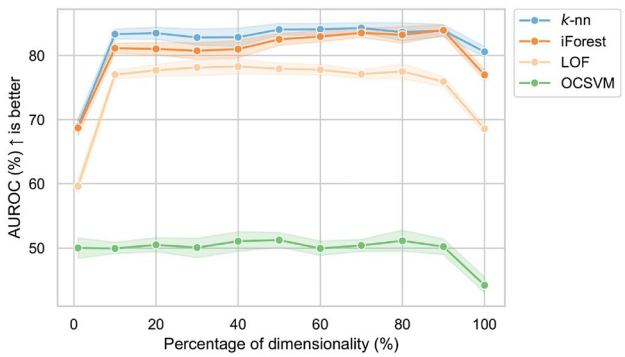


Fig. 9 Ablation study showing how shallow detector results vary with subspace dimensionality

performance generally remains stable when discarding more components - up to the top 90%.

This observation aligns with previous findings that show residual directions capture information important for out-of-distribution detection [25]. The magnitude of normal data is minute in this space, which is not necessarily the case for anomalies. Based on these results, we do not need complete neural network representations to perform anomaly detection. A subset suffices.

4.5 Synthetic anomalies

Anomaly detection depends on two factors: the nature of the normal data and the nature of anomalies. Both classes can originate from complex, irregular distributions. These aspects make it difficult to pinpoint the causes of results on ODDS and other curated datasets. We attempt to disentangle these factors by analysing performance on synthetic anomalies. The anomalies curated in ODDS are a composite of these types. We calculated the correlation between the ODDS and the synthetic anomaly scores and found that the datasets exhibited correlations between multiple synthetic categories, highlighting the complex qualities of the anomalies. For example, when analysing the raw data representations, k-NN on the curated *Letter* anomalies correlates strongly with local ($\rho = 0.84$), global ($\rho = 0.49$), and dependency ($\rho = 0.94$) anomaly scores.

Figure 10a–d shows the results across the four synthetic types. We show comparisons using k-NN as we found similar behaviours across the detectors. The contrastive objectives outperform the baseline in the local (Fig. 10a) and cluster anomaly (Fig. 10b) scenarios. This result suggests contrastive tasks are better at discerning differences at a local neighbourhood level.

No self-supervised approach beats the baseline when faced with global anomalies (Fig. 10c). This result contributes to the idea that self-supervised representations introduce irrelevant directions. Since the global anomalies scatter across the representation space, these additional directions mask the meaningful distances between the anomalies and normal points. As a result, methods like k-NN become less effective. In addition, the ranking of the self-supervised tasks aligns most closely with their rankings on ODDS (Fig. 13), which potentially highlights the overall properties of the ODDS datasets.

For the dependency anomalies, rotation and mask classification surpass the baseline (Fig. 10d). Conversely, contrastive tasks perform the worst. Using a rotation or mask classification pretext task could help promote the intrinsic property that tabular data are non-invariant, which may help identify this type of anomaly.

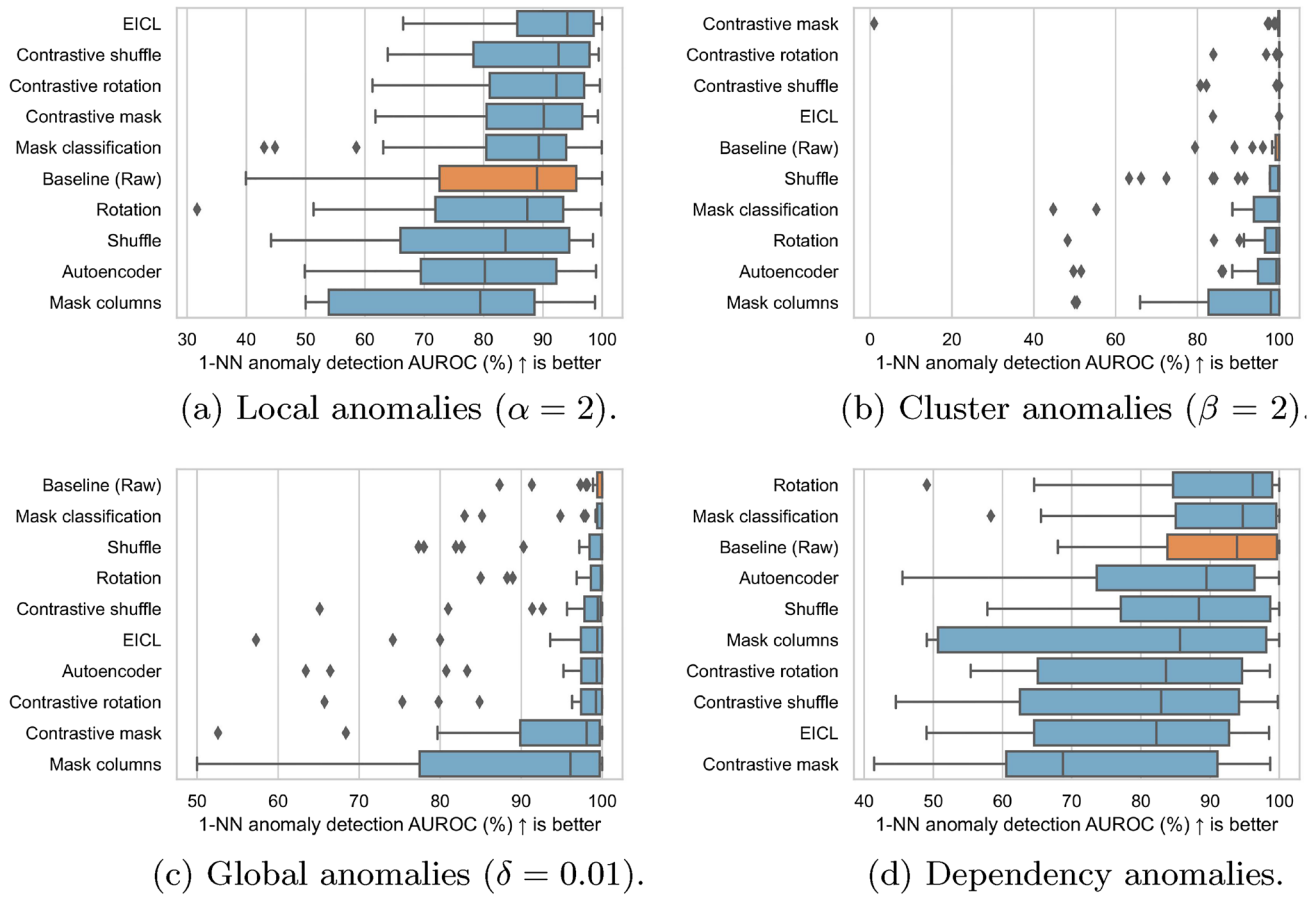


Fig. 10 Bar plots comparing synthetic anomaly results across the representations

4.6 Architectural choices for self-supervision

We analyse the effects of architectures and loss functions on performance to provide starting points for improving deep learning methods for tabular anomaly detection. We illustrate the results using k-NN as we observe similar behaviours across detectors.

ResNets outperform transformers. Our experiments indicate ResNets are a better choice than FT-Transformer (Fig. 11a). This result may be due to transformers needing more training data during the learning phase [52]—the ODDS datasets are relatively small.

Standardisation is not necessary. Standardising data before training neural networks does not offer much benefit (Fig. 11b).

ARPL is a better choice for classification-type losses. ARPL significantly outperforms cross-entropy and AAM when training classification-type tasks (Fig. 11c). Specialised losses like ARPL might represent "other" spaces better in the context of smaller datasets.

InfoNCE is better than VICReg for contrastive-type losses. This result (Fig. 11d) may be due to the intricacies of VICReg, which requires balancing three components (pair similarity, variance and covariance).

4.7 Benchmarking unsupervised anomaly detection

Finally, we compare the performance of each of the detectors overall to see how well they perform in one-class settings. We aggregate results across the baseline and self-supervised embeddings to provide a more generalised understanding of detector behaviour.

Figures 12 and 13 summarise the overall performances of each anomaly detector on ODDS. Even with the inclusion of self-supervised representations, k-NN performs best. Our findings align with other works highlighting k-NN as a performant anomaly detector [12–14, 17]. However, apart from k-NN and residual norm, Fig. 13 shows no significant statistical differences between the detectors, suggesting the detectors make similar classification decisions. k-NN might be a sensible starting point that does not make strong assumptions about the normal distribution. Nonetheless, the choice

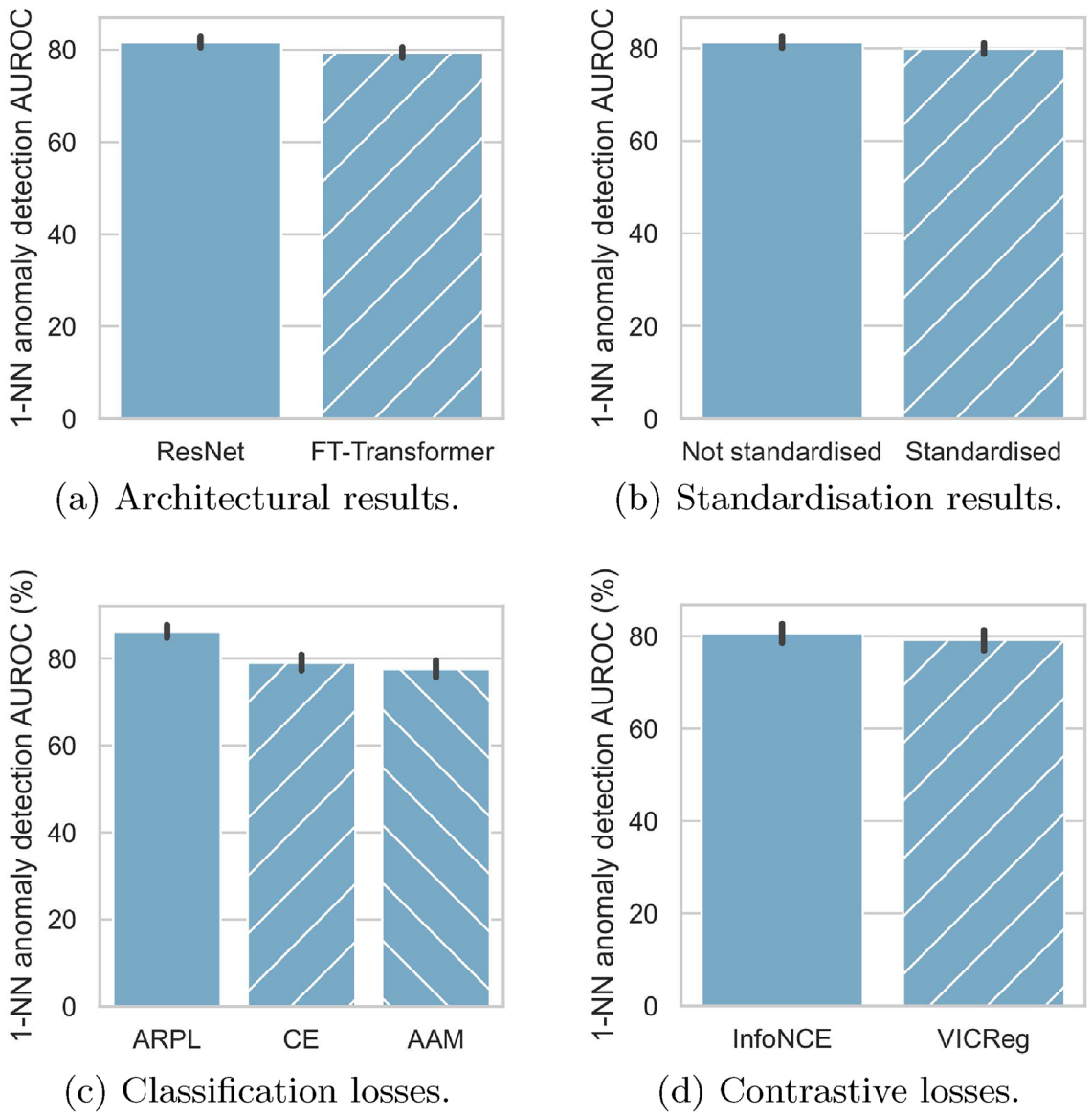


Fig. 11 Comparisons of how architecture and losses affect performance on the self-supervised embeddings

of underlying representation should take precedence over the detector when designing anomaly detection systems.

4.7.1 Hyperparameter ablations

We now examine the sensitivity of the detectors to changes in hyperparameters. These experiments were conducted directly on the raw ODDS data only to understand detector performance in an optimal representation space. By doing so, these results enable a better understanding of the detectors' inductive biases and why they may deteriorate in sub-optimal self-supervised representations.

***k*-NN**: Figure 14 shows performance remains relatively stable to changes in k, suggesting the choice of this hyperparameter is trivial. As k-NN considers global relationships, this result indicates that anomalies already lie in distinct regions separate from the normal raw data.

LOF: Figure 15 illustrates how LOF performance changes with k. Although LOF and k-NN consider points in a neighbourhood, LOF is more sensitive to the number of neighbours (as evidenced by the increase in performance when $k = 1$ and $k = 5$ for LOF). However, it is unclear how to choose a value of k so that LOF is competitive with the other detectors in the one-class setting.

Residual norms: Figure 16 shows how performance varies with the percentage of attributes used. There are no notable trends, although performance remains better than random, even with a small subset (10%) of features. The

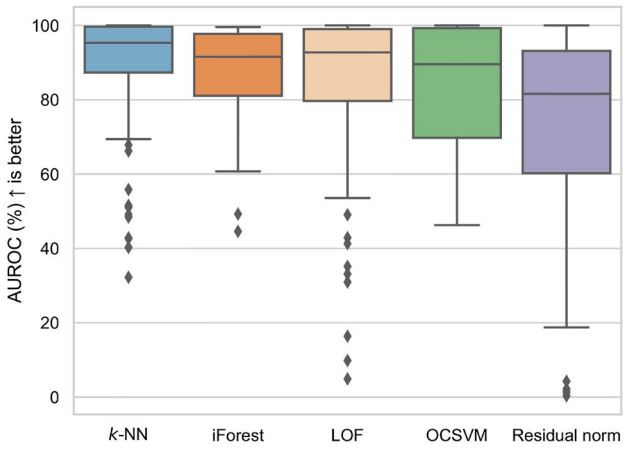


Fig. 12 Box plot comparing detector performance on the raw and standardised data. The results include all hyperparameter variations where available

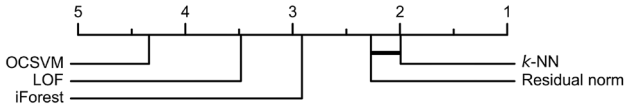


Fig. 13 Critical difference diagram ranking the different detectors. The dark lines between different detectors indicate a statistical difference ($p < 0.05$) in results when running pairwise comparison tests

number of relevant attributes in the original representation space is dataset-dependent as ODDS contains datasets from differing tasks. It is unclear how to choose the number of features to maximise the performance of residual norms in the original dataset space.

4.7.2 Corrupted input data

Adding uninformative features: All detectors are sensitive to irrelevant features (Fig. 17a). Although residual norms do not achieve the highest performance, it is more stable under increasing noise levels. This result may be due to the residuals capturing the most meaningful directions of the data. In contrast, *k*-NN performance declines the most.

Removing and selecting important features: Overall, performance plateaus at around 50% of attributes, suggesting half of the raw features are irrelevant for anomaly detection. iForest and OCSVM are the most stable under varying subsets of features (Figs. 17b and c).

Missing values: Most detectors exhibit a slight decline in AUROC with increasing proportions of missing values

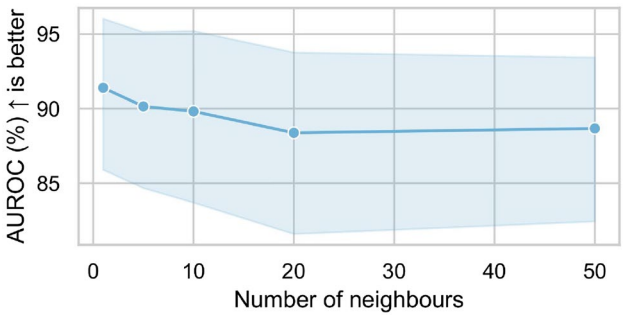


Fig. 14 Line plot showing how *k*-NN varies with the change in the number of nearest neighbours, aggregated across the ODDS datasets, with 95% confidence intervals

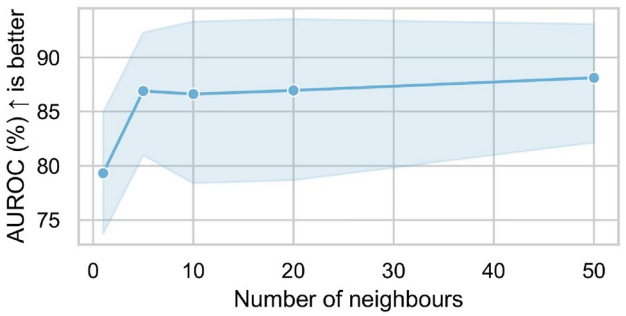


Fig. 15 Line plot showing how LOF varies with the change in the number of nearest neighbours, aggregated across the ODDS dataset, with 95% confidence intervals

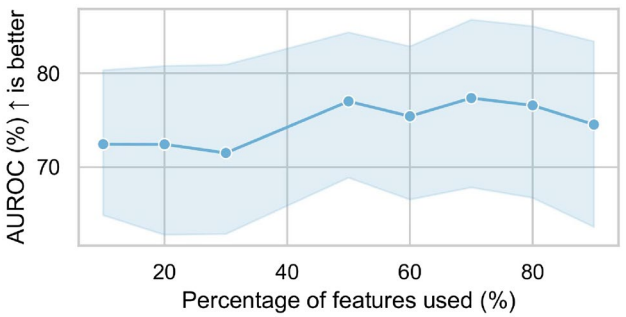


Fig. 16 Line plot showing how residual norm varies with the change in residual dimensionality, aggregated across the ODDS dataset, with 95% confidence intervals

(Fig. 17d). LOF is the exception, as performance drops significantly.

Overall, the results indicate *k*-NN is the best-performing detector when faced with clean and relevant features. However, the relative ranking of detectors changes in the

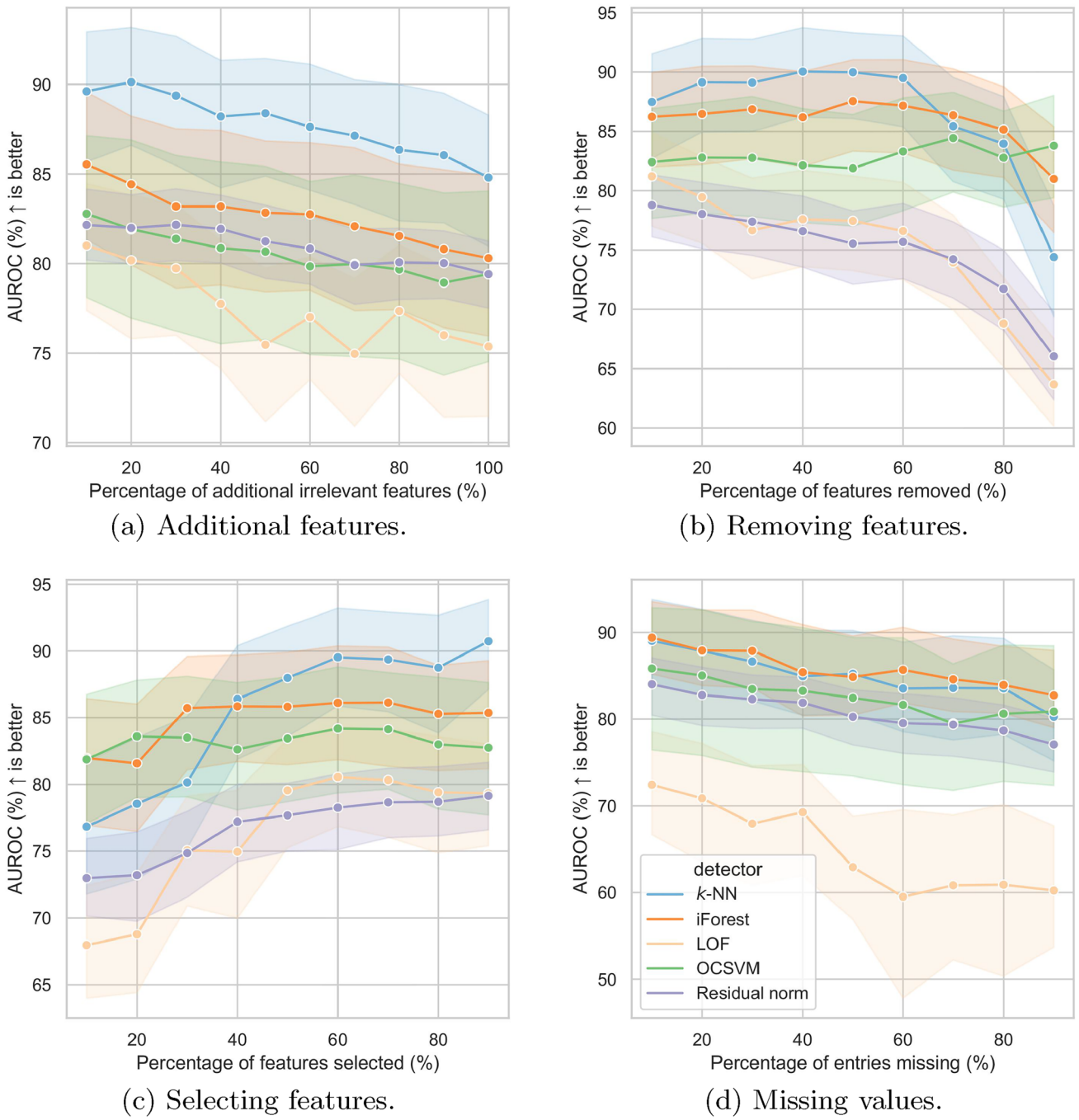


Fig. 17 Ablations showing how detector performance varies with changing levels of corrupt data

presence of corrupted input data. As observed in our self-supervised results (Sect. 4.4), residual norms might be better at filtering out noisy directions. Furthermore, when there are fewer relevant features, iForest may be a better choice.

5 Conclusion

5.1 Limitations and future work

We limited our experiments to the ODDS, which is not necessarily representative of all tabular anomaly datasets. Several datasets underwent preprocessing during the curation of ODDS, which could affect results. For example, the values

in *HTTP* were log-transformed. In addition, the datasets are relatively small. As neural networks (particularly transformers) benefit from large amounts of data [52], it is unclear if self-supervision would be more advantageous in the big data case. Contrastive objectives are particularly reliant on large datasets and batch sizes [30, 31]. Additional ablations could examine the effects of dataset size on representation quality and detection performance.

Furthermore, we isolated our analyses by extracting embeddings at the penultimate layer and running shallow anomaly detection algorithms. Although feature extraction at this stage combined with simple detectors is a popular strategy [12–14, 17], different parts of the neural network could provide more informative features [22]. Moreover, we chose to use shallow detectors to prioritise studying the effect of representations rather than studying the detection approach. In addition, the original implementations of ICL and GOAD evaluate anomalies using an entire neural network pipeline and use specific architectures for the tasks. Adapting these implementations for a pretext task with different architectures deviates from the original setup and could affect performance. Future work could look at extending the experiments to examine how varying pretext tasks with deep anomaly detection can yield better results [53].

Another direction for future work that focuses on representation quality could replace the one-class detectors with semi-supervised or supervised classifiers. We decided to concentrate on one-class detectors to align with the anomaly detection field [3, 10, 15, 16]. However, anomalies can manifest in different ways, and it could be challenging for an unsupervised detector to capture the relevant features for a specific task in practice. Incorporating prior knowledge about anomalies through weak or semi-supervised detection approaches could improve detection [54].

In addition, studies focusing on improving deep tabular anomaly detectors could also start examining regularisation strategies. Our experiments suggest neural networks add irrelevant features; hence, regularisation during the training process could help to control this behaviour.

5.2 Summary

We trained multiple neural networks on various self-supervised pretext tasks to learn new representations for ODDS, a series of tabular anomaly detection datasets. We ran a suite of shallow anomaly detectors on the new embeddings and compared the results to the performance of the original data. None of the self-supervised representations outperformed the raw baseline.

We conducted ablations to try to understand this behaviour. Our empirical findings suggested that neural networks introduce irrelevant features, which degrade detector capability. As normal and anomalous data were easily distinguishable in the original tabular representations, neural networks merely stretched the data. They did not introduce any additional informative information. However, we demonstrated performance was recoverable by projecting the embeddings to a residual subspace.

As the anomalies from ODDS derive from complex distributions, we repeated the experiments on synthetic data to understand the pretext tasks' influence on detecting particular anomaly types. We showed in specific scenarios that self-supervision can be beneficial. Contrastive tasks were better at picking up localised anomalies, while classification tasks were better at identifying differences in dependency structures.

Finally, we studied different shallow detectors by aggregating performances across the baseline and self-supervised representations. We showed that localised methods like k-NN and LOF worked best on ODDS but were susceptible to performance degradation with corrupted data. In contrast, iForest was more robust. Our findings provided practical insights into when one detector might be preferable to another.

Overall, our findings complement the growing landscape of theories on why self-supervised learning works. Effective self-supervised pretext tasks learn to compress the input data when there are irrelevant features [55–57]. Our findings suggest current deep learning approaches do not add much benefit when the original feature space succinctly represents the normal data. This situation is often the case for tabular data, and we demonstrated this by showing performance degrades when removing features in the original space. If the feature space did not succinctly represent the normal data, we would not observe such large degradations. This setup differs from other domains. For example, pixels in images contain lots of semantically irrelevant information. Therefore, neural networks can distil information from pixels to extract useful semantic features and self-supervision is beneficial.

Appendix

See Figs. 18 and 19.

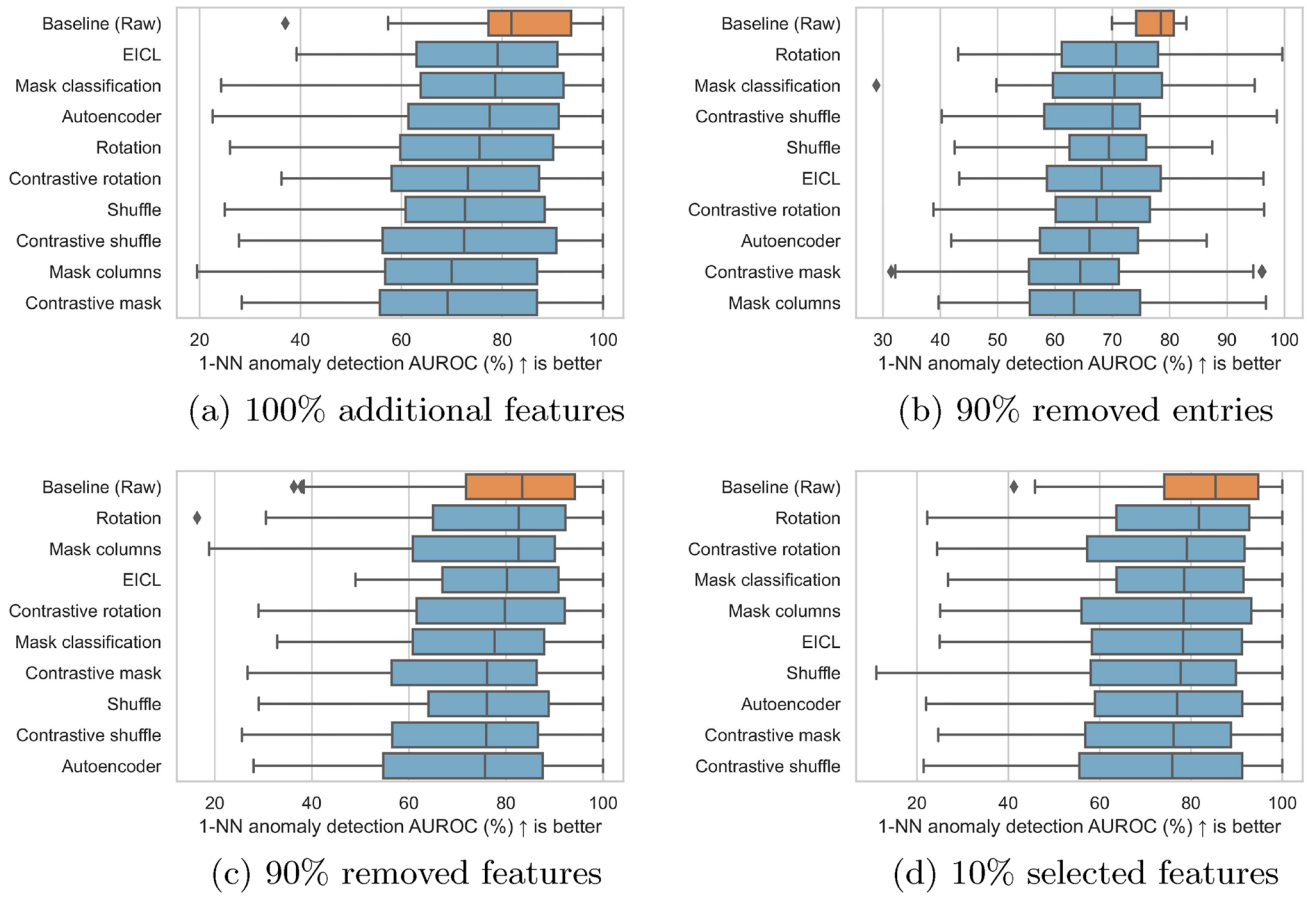


Fig. 18 Box plot comparing nearest neighbour AUROCs for each of the self-supervised pretext tasks on corrupted input data

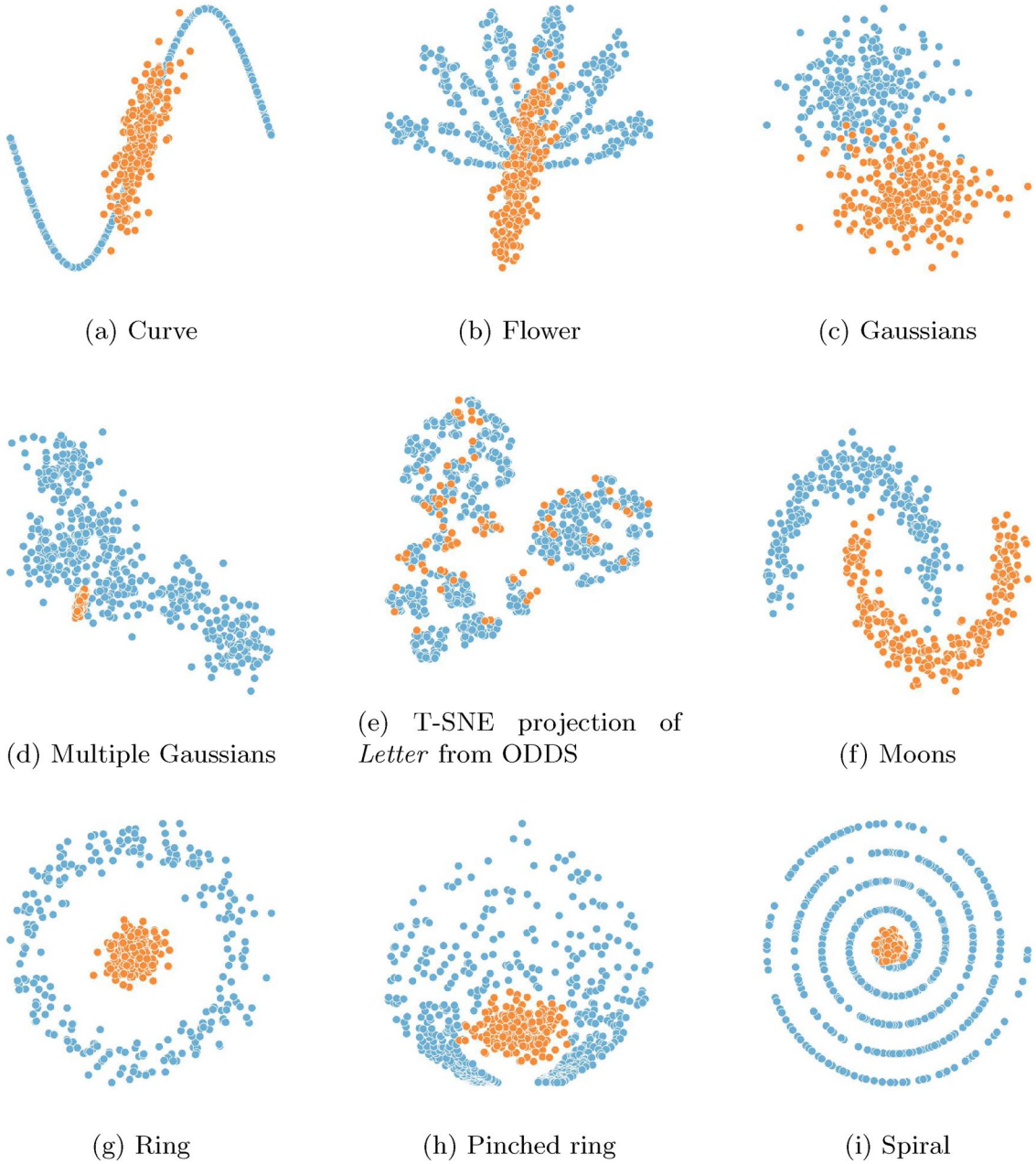


Fig. 19 Illustrations of the toy test data. Blue points are normal whereas orange points are anomalous

Author contributions KTM, TD, and LDG contributed to Conceptualisation; KTM contributed to Formal analysis, Investigation, Visualisation, Writing—original draft and Software; KTM and LDG contributed to Methodology; and TD and LDG contributed to Supervision and Writing—review & editing.

Funding KTM is supported by EPSRC under Grant EP/R513143/1. This work was supported by funding from EPSRC under Grant EP/R513143/1.

Data availability Publicly available datasets were analysed in this study. The ODDS datasets are accessible from https://odds.cs.stonybrook.edu/.

Declarations

Competing interests The authors declare no competing interests.

Ethical approval Not applicable.

References

1. Hendrycks D, Mazeika M, Kadavath S, Song D (2019) Using self-supervised learning can improve model robustness and uncertainty. Curran Associates, Red Hook
2. Mai KT, Davies T, Griffin LD (2022) Self-supervised losses for one-class textual anomaly detection. CoRR https://doi.org/10.48550/arXiv.2204.05695
3. Reiss T, Cohen N, Horwitz E, Abutbul R, Hoshen Y (2022) Anomaly detection requires better representations. In: Karlinsky L, Michaeli T, Nishino K (eds) Computer vision - ECCV 2022 workshops, October 23-27, 2022, Proceedings, Part IV. Lecture Notes in Computer Science, vol 13804. Springer, Tel Aviv, Israel, pp 56–68. https://doi.org/10.1007/978-3-031-25069-9_4
4. Balestriero R, Ibrahim M, Sobal V, Morcos A, Shekhar S, Goldstein T, Bordes F, Bardes A, Mialon G, Tian Y, Schwarzschild A, Wilson AG, Geiping J, Garrido Q, Fernandez P, Bar A, Pirsiavash H, LeCun Y, Goldblum M (2023) A cookbook of self-supervised learning. CoRR https://doi.org/10.48550/arXiv.2304.12210
5. Zhang R, Isola P, Efros AA (2016) Colorful image colorization. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer Vision - ECCV 2016 - 14th European Conference, October 11-14, 2016, Proceedings, Part III. Lecture Notes in Computer Science, vol 9907. Springer, Amsterdam, The Netherlands, pp 649–666. https://doi.org/10.1007/978-3-319-46487-9_40
6. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Walker MA, Ji H, Stent A (eds) Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2018, June 1-6, 2018, Volume 1 (Long Papers). Association for Computational Linguistics, New Orleans, Louisiana, USA, pp 2227–2237. https://doi.org/10.18653/v1/n18-1202
7. Radford A, Narasimhan K, Salimans T, Sutskever I et al (2018) Improving language understanding by generative pre-training
8. Geirhos R, Rubisch P, Michaelis C, Bethge M, Wichmann FA, Brendel W (2019) Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In: 7th International Conference on Learning Representations, ICLR 2019, May 6-9, 2019. OpenReview.net, New Orleans, Louisiana, USA. https://openreview.net/forum?id=Bygh9j09KX
9. Hermann KL, Chen T, Kornblith S (2020) The origins and prevalence of texture bias in convolutional neural networks. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H (eds) Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual. https://proceedings.neurips.cc/paper/2020/hash/db5f9f42a7157abe65bb145000b5871a-Abstract.html
10. Ruff L, Kauffmann JR, Vandermeulen RA, Montavon G, Samek W, Kloft M, Dietterich TG, Müller K (2021) A unifying review of deep and shallow anomaly detection. Proc IEEE 109(5):756–795. https://doi.org/10.1109/JPROC.2021.3052449
11. Tack J, Mo S, Jeong J, Shin J (2020) CSI: novelty detection via contrastive learning on distributionally shifted instances. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H (eds) Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual. https://proceedings.neurips.cc/paper/2020/hash/8965f76632d7672e7d3cf29c87ecaa0c-Abstract.html
12. Sehwag V, Chiang M, Mittal P (2021) SSD: a unified framework for self-supervised outlier detection. In: 9th International Conference on Learning Representations, ICLR 2021, May 3-7, 2021. OpenReview.net, Virtual. https://openreview.net/forum?id=v5gjXpmR8J
13. Reiss T, Cohen N, Bergman L, Hoshen Y (2021) PANDA: adapting pretrained features for anomaly detection and segmentation. In: IEEE conference on computer vision and pattern recognition, CVPR 2021, virtual, June 19-25, 2021, pp 2806–2814. Computer Vision Foundation/IEEE, Virtual. https://doi.org/10.1109/CVPR46437.2021.00283 . https://openaccess.thecvf.com/content/CVPR2021/html/Reiss_PANDA_Adapting_Pretrained_Features_for_Anomaly_Detection_and_Segmentation_CVPR_2021_paper.html
14. Sun Y, Ming Y, Zhu X, Li Y (2022) Out-of-distribution detection with deep nearest neighbors. In: Chaudhuri K, Jegelka S, Song L, Szepesvári C, Niu G, Sabato S (eds) International conference on machine learning, ICML 2022, 17-23 July 2022. Proceedings of machine learning research, vol 162, pp 20827–20840. PMLR, Baltimore, Maryland, USA. https://proceedings.mlr.press/v162/sun22d.html
15. Schölkopf B, Williamson RC, Smola AJ, Shawe-Taylor J, Platt JC (1999) Support vector method for novelty detection. In: Solla SA, Leen TK, Müller K (eds) Advances in neural information processing systems 12, [NIPS Conference, November 29 - December 4, 1999], pp 582–588. The MIT Press, Denver, Colorado, USA. http://papers.nips.cc/paper/1723-support-vector-method-for-novelty-detection
16. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3):15–11558. https://doi.org/10.1145/1541880.1541882
17. Gu X, Akoglu L, Rinaldo A (2019) Statistical analysis of nearest neighbor methods for anomaly detection. In: Wallach HM, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox EB, Garnett R (eds) Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp 10921–10931. https://proceedings.neurips.cc/paper/2019/hash/805163a0f0f128e473726ccda5f91bac-Abstract.html
18. Bergman L, Cohen N, Hoshen Y (2020) Deep nearest neighbor anomaly detection. CoRR arXiv: 2002.10445
19. Breunig MM, Kriegel H, Ng RT, Sander J (2000) LOF: identifying density-based local outliers. In: Chen W, Naughton JF, Bernstein PA (eds) Proceedings of the 2000 ACM SIGMOD international conference on management of data, May 16-18, 2000, Dallas, Texas, USA. ACM, Dallas, Texas, USA, pp 93–104. https://doi.org/10.1145/342009.335388
20. Liu FT, Ting KM, Zhou Z (2008) Isolation forest. In: Proceedings of the 8th IEEE international conference on data mining (ICDM 2008), December 15-19, 2008. IEEE Computer Society, Pisa, Italy, pp 413–422. https://doi.org/10.1109/ICDM.2008.17

21. Huang L, Nguyen X, Garofalakis MN, Jordan MI, Joseph AD, Taft N (2006) In-network PCA and anomaly detection. In: Schölkopf B, Platt JC, Hofmann T (eds) Advances in neural information processing systems 19, proceedings of the twentieth annual conference on neural information processing systems, December 4-7, 2006. MIT Press, Vancouver, British Columbia, Canada, pp 617–624. https://proceedings.neurips.cc/paper/2006/hash/2227d753dc18505031869d44673728e2-Abstract.html
22. Kim KH, Shim S, Lim Y, Jeon J, Choi J, Kim B, Yoon AS (2020) Rapp: Novelty detection with reconstruction along projection pathway. In: 8th International conference on learning representations, ICLR 2020, April 26-30, 2020. OpenReview.net, Addis Ababa, Ethiopia. https://openreview.net/forum?id=HkgeGeBYDB
23. Wang H, Li Z, Feng L, Zhang W (2022) Vim: Out-of-distribution with virtual-logit matching. In: IEEE/CVF conference on computer vision and pattern recognition, CVPR 2022, June 18-24, 2022. IEEE, New Orleans, Louisiana, USA, pp 4911–4920. https://doi.org/10.1109/CVPR52688.2022.00487
24. Yang J, Wang P, Zou D, Zhou Z, Ding K, Peng W, Wang H, Chen G, Li B, Sun Y, Du X, Zhou K, Zhang W, Hendrycks D, Li Y, Liu Z (2022) Openood: Benchmarking generalized out-of-distribution detection. In: NeurIPS. http://papers.nips.cc/paper_files/paper/2022/hash/d201587e3a84fc4761eadc743e9b3f35-Abstract-Datasets_and_Benchmarks.html
25. Kamoi R, Kobayashi K (2020) Why is the mahalanobis distance effective for anomaly detection? CoRR arXiv: 2003.00402
26. Gidaris S, Singh P, Komodakis N (2018) Unsupervised representation learning by predicting image rotations. In: 6th International conference on learning representations, ICLR 2018, April 30 - May 3, 2018, conference track proceedings. OpenReview.net, Vancouver, British Columbia, Canada. https://openreview.net/forum?id=S1v4N2l0-
27. Noroozi M, Favaro P (2016) Unsupervised learning of visual representations by solving jigsaw puzzles. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer Vision - ECCV 2016 - 14th European conference, October 11-14, 2016, Proceedings, Part VI. Lecture Notes in Computer Science, vol. 9910. Springer, Amsterdam, The Netherlands, pp 69–84. https://doi.org/10.1007/978-3-319-46466-4_5
28. Pathak D, Krähenbühl P, Donahue J, Darrell T, Efros AA (2016) Context encoders: Feature learning by inpainting. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, June 27-30, 2016. IEEE Computer Society, Las Vegas, Nevada, USA, pp 2536–2544. https://doi.org/10.1109/CVPR.2016.278
29. He K, Chen X, Xie S, Li Y, Dollár P, Girshick RB (2022) Masked autoencoders are scalable vision learners. In: IEEE/CVF conference on computer vision and pattern recognition, CVPR 2022, June 18-24, 2022. IEEE, pp 15979–15988. https://doi.org/10.1109/CVPR52688.2022.01553
30. Oord A, Li Y, Vinyals O (2018) Representation learning with contrastive predictive coding. CoRR arXiv: 1807.03748
31. Chen T, Kornblith S, Norouzi M, Hinton GE (2020) A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th international conference on machine learning, ICML 2020, 13-18 July 2020. Proceedings of Machine Learning Research, vol. 119, pp 1597–1607. PMLR, Virtual (2020). http://proceedings.mlr.press/v119/chen20j.html
32. Dwibedi D, Aytar Y, Tompson J, Sermanet P, Zisserman A (2021) With a little help from my friends: nearest-neighbor contrastive learning of visual representations. In: 2021 IEEE/CVF international conference on computer vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. IEEE, pp 9568–9577. https://doi.org/10.1109/ICCV48922.2021.00945
33. Biswas M, Buckchash H, Prasad DK (2023) pnnclr: Stochastic pseudo neighborhoods for contrastive learning based unsupervised representation learning problems. CoRR https://doi.org/10.48550/ARXIV.2308.06983
34. Bardes A, Ponce J, LeCun Y (2022) Vicreg: variance-invariance-covariance regularization for self-supervised learning. In: The tenth international conference on learning representations, ICLR 2022, April 25-29, 2022. OpenReview.net, Virtual. https://openreview.net/forum?id=xm6YD62D1Ub
35. Golan I, El-Yaniv R (2018) Deep anomaly detection using geometric transformations. In: Bengio S, Wallach HM, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R (eds) Advances in neural information processing systems 31: annual conference on neural information processing systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pp 9781–9791. https://proceedings.neurips.cc/paper/2018/hash/5e62d03aec0d17facfc5355dd90d441c-Abstract.html
36. Bergman L, Hoshen Y (2020) Classification-based anomaly detection for general data. In: 8th international conference on learning representations, ICLR 2020, April 26-30, 2020. OpenReview.net, Addis Ababa, Ethiopia (2020). https://openreview.net/forum?id=H1lK_lBtvS
37. Shenkar T, Wolf L (2022) Anomaly detection for tabular data with internal contrastive learning. In: The tenth international conference on learning representations, ICLR 2022 April 25-29, 2022. OpenReview.net, Virtual. https://openreview.net/forum?id=_hszZbt46bT
38. Rayana S (2016) ODDS Library. https://odds.cs.stonybrook.edu
39. Han S, Hu X, Huang H, Jiang M, Zhao Y (2022) Adbench: Anomaly detection benchmark. In: NeurIPS. http://papers.nips.cc/paper_files/paper/2022/hash/cf93972b116ca5268827d575f2cc226b-Abstract-Datasets_and_Benchmarks.html
40. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, VanderPlas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830. https://doi.org/10.5555/1953048.2078195
41. Johnson J, Douze M, Jégou H (2021) Billion-scale similarity search with gpus. IEEE Trans Big Data 7(3):535–547. https://doi.org/10.1109/TBDATA.2019.2921572
42. Grinsztajn L, Oyallon E, Varoquaux G (2022) Why do tree-based models still outperform deep learning on typical tabular data? In: NeurIPS. http://papers.nips.cc/paper_files/paper/2022/hash/0378c7692da36807bdec87ab043cdadc-Abstract-Datasets_and_Benchmarks.html
43. Yoon J, Zhang Y, Jordon J, Schaar M (2020) VIME: extending the success of self- and semi-supervised learning to tabular domain. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H (eds) Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. https://proceedings.neurips.cc/paper/2020/hash/7d97667a3e056acab9aaf653807b4a03-Abstract.html
44. Gorishniy Y, Rubachev I, Khrulkov V, Babenko A (2021) Revisiting deep learning models for tabular data. In: Ranzato M, Beygelzimer A, Dauphin YN, Liang P, Vaughan JW (eds) Advances in neural information processing systems 34: annual conference on neural information processing systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pp 18932–18943. https://proceedings.neurips.cc/paper/2021/hash/9d86d83f925f2149e9edb0ac3b49229c-Abstract.html
45. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, June 27-30, 2016. IEEE

Computer Society, Las Vegas, Nevada, USA, pp 770–778. https://doi.org/10.1109/CVPR.2016.90
46. Chen G, Peng P, Wang X, Tian Y (2022) Adversarial reciprocal points learning for open set recognition. IEEE Trans Pattern Anal Mach Intell 44(11):8065–8081. https://doi.org/10.1109/TPAMI.2021.3106743
47. Vaze S, Han K, Vedaldi A, Zisserman A (2022) Open-set recognition: a good closed-set classifier is all you need. In: The tenth international conference on learning representations, ICLR 2022, April 25-29, 2022. OpenReview.net, virtual. https://openreview.net/forum?id=5hLP5JY9S2d
48. Deng J, Guo J, Yang J, Xue N, Kotsia I, Zafeiriou S (2022) Arcface: additive angular margin loss for deep face recognition. IEEE Trans Pattern Anal Mach Intell 44(10):5962–5979. https://doi.org/10.1109/TPAMI.2021.3087709
49. Ming Y, Sun Y, Dia O, Li Y (2023) How to exploit hyperspherical embeddings for out-of-distribution detection? In: The eleventh international conference on learning representations https://openreview.net/forum?id=aEFaE0W5pAd
50. Steinbuss G, Böhm K (2021) Benchmarking unsupervised outlier detection with realistic synthetic data. ACM Trans Knowl Discov Data 15(4):65–16520. https://doi.org/10.1145/3441453
51. Stolfo S, Fan W, Lee W, Prodromidis A, Chan P (1999) KDD Cup 1999 Data. UCI Machine Learning Repository. https://doi.org/10.24432/C51C7N
52. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2021) An image is worth 16x16 words: Transformers for image recognition at scale. In: 9th International conference on learning representations, ICLR 2021, May 3-7, 2021. OpenReview.net, virtual. https://openreview.net/forum?id=YicbFdNTTy
53. Ruff L, Görnitz N, Deecke L, Siddiqui SA, Vandermeulen RA, Binder A, Müller E, Kloft M (2018) Deep one-class classification. In: Dy JG, Krause A (eds) Proceedings of the 35th international conference on machine learning, ICML 2018, July 10–15, 2018. Proceedings of machine learning research, vol 80. PMLR, Stockholmsmässan, Stockholm, Sweden, pp 4390–4399. http://proceedings.mlr.press/v80/ruff18a.html
54. Ruff L, Vandermeulen RA, Görnitz N, Binder A, Müller E, Müller K, Kloft M (2020) Deep semi-supervised anomaly detection. In: 8th International conference on learning representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net. https://openreview.net/forum?id=HkgH0TEYwH
55. Lee K, Arnab A, Guadarrama S, Canny JF, Fischer I (2021) Compressive visual representations. In: Ranzato M, Beygelzimer A, Dauphin YN, Liang P, Vaughan JW (eds) Advances in neural information processing systems 34: annual conference on neural information processing systems 2021, NeurIPS 2021, December 6-14, 2021, Virtual, pp 19538–19552. https://proceedings.neurips.cc/paper/2021/hash/a29a5ba2cb7bdeabba22de8c83321b46-Abstract.html
56. Shwartz-Ziv R, LeCun Y (2023) To compress or not to compress - self-supervised learning and information theory: a review. CoRR https://doi.org/10.48550/ARXIV.2304.09355
57. Yu Y, Buchanan S, Pai D, Chu T, Wu Z, Tong S, Bai H, Zhai Y, Haeffele BD, Ma Y (2023) White-box transformers via sparse rate reduction: compression is all there is? https://doi.org/10.48550/arXiv.2306.01129