Representation Learning for Anomaly Detection in Different Modalities

Kimberly Thien Ton-Mai

A dissertation submitted in partial fulfilment

of the requirements for the degree of

Doctor of Philosophy

of

University College London

Department of Security and Crime Science

University College London

December 2024

I confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Kimberly Thien Ton-Mai

December 2024

Acknowledgements

This endeavour would not have been possible without my supervisors, Professor Lewis Griffin and Dr Toby Davies. I am grateful they were willing to take me on when I finished my master's degree and I had no idea what to do, and for always making time to discuss my ideas.

I sincerely appreciate the Dawes Centre for Future Crime for introducing me to the world of policy and helping me take the next step in my career journey after my PhD.

Finally, I thank my parents and Kelvin for their patience and support during my studies.

Abstract

Anomaly detection is the task of identifying unusual instances that deviate from typical appearances or behaviours. Use cases include fraud detection, medicine, and fault detection. Effective automated systems can identify anomalies in situations that are too challenging or costly for humans. However, satisfactory detection performance relies on underlying representation space that depicts the training data. In this thesis, we investigate what characteristics form a good representation. We conduct experiments on images, text, speech, and tabular data to examine how well anomalies can be detected in each case and to find commonalities across the modalities.

We find that no representation learning scheme performs well across all modalities. However, our results suggest that low-dimensional embeddings are best for anomaly detection. Using embeddings from pre-trained networks is an effective starting point and fine-tuning boosts performance. We also analyse how the anomaly detection architectures affect results. We show the detector is unimportant as long as the representation space is reasonable. The choice of representation should consider prior knowledge about the anomalies and how they contrast with the benign distribution. Overall, our findings suggest anomaly detection research should focus on representation learning objectives rather than modifying architectures or scoring functions.

Impact statement

The need for automated safety systems is becoming more pronounced in a world of continuous technological innovation. As technology moves at lightning speed, the data supporting this technology becomes more complex. The increasing scale and complexity of this data are overwhelming for humans. This creates a need for automated safety systems. These systems are necessary for detecting accidental faults and to guard against the actions of increasingly sophisticated adversaries. In many cases, safety-critical tasks are equivalent to anomaly detection problems. These tasks range from locating firearms in airport luggage to identifying disinformation.

We demonstrate how difficult these safety-critical tasks are for humans. We conducted a large-scale study to measure how well humans could detect speech deepfakes. This was the first study to examine capability in English and Mandarin as we recognised that deepfakes are a global problem. We found humans cannot reliably detect speech deepfakes even when primed for the task. Our work resonated with the public and received coverage across international media, including The Guardian, New Scientist, and the BBC. Following this, we have continuously engaged with the media to explain how AI can impact daily lives. We have fact-checked disinformation, explained how deepfakes work on podcasts, and created spoofs for a radio programme on AI-enabled fraud.

Although automated systems are imperative for decision-making, the work shows that supervised classifiers are insufficient. They do not generalise well to new test conditions, such as different speakers. Our findings signal the need for alternative approaches, especially as issues like non-consensual deepfake proliferation increasingly affect society.

This thesis examines the capabilities of anomaly detectors, which do not need labelled anomalies for training. We study their performance across different modalities to find potential synergies. The results show that the choice of representation is the most crucial component in an anomaly detector. Simple detectors work, provided the underlying benign embedding is low-dimensional. This property is achievable by adapting pre-trained neural networks trained on diverse data. We also provide instances where particular detectors and architectures work better than others.

The findings will interest practitioners who want to deploy anomaly detectors in production to address safety-critical problems. Our work signals promising directions and areas to disregard in anomaly detection research. We establish a time and environmentally-efficient baseline for anomaly detection by showing that a detector with pre-trained embeddings performs reasonably.

Future work could build on the thesis by curating training datasets to improve pre-trained neural networks or testing the resilience of anomaly detectors in more challenging settings, such as corrupted input data.

Contents

List of Figures

List of Tables

Acronyms

GMM gaussian mixture model. 16, 91

GNN graph neural network. 81

iForest isolation forest. 116

ITW In-The-Wild. 88, 107, 108, 111

KL Kullback-Leibler. 41

LA logical access. 86

LFCC linear frequency cepstral coefficients. 65, 89, 100, 106, 109

LLM large language model. 42

LOF local outlier factor. 116

MAE mean absolute error. 119

MDF Mahalanobis distance as features. 41, 45, 52, 55

MFCC Mel frequency cepstral coefficients. 80, 89, 106

MLM masked language modelling. 43, 45, 47–50, 53, 54

MLP multilayer perceptron. 82

MSE mean squared error. 22–24, 31, 35, 119

NLP natural language processing. 38

OCSVM one-class support vector machine. 16, 17, 41, 45, 51, 82, 113, 116

OOD out-of-distribution. 7, 8, 16, 17, 19, 33, 38–42, 44, 46, 48, 78, 82, 90, 95

RMD replaced mask detection. 42

RNN recurrent neural network. 38

RTD replaced token detection. 42

SoC stream-of-commerce. 25

SSAST self-supervised audio spectrogram transformer. 89, 92

SVDD support vector data description. 40

TTS text-to-speech. 59, 62, 87, 88, 101, 110, 141

VC voice conversion. 87, 101, 110, 141

VCTK Voice Cloning Toolkit. 86, 87

VIT vision transformer. 89, 92, 107, 109

1 | Introduction

Malicious actors are constantly innovating in this era of rapid technological advancement. As security systems learn to counter the modus operandi of criminals, new strategies to bypass defences emerge. One example is identity fraud. Previously, a fraudster might impersonate a victim using a crude mask to mimic their appearance and mannerisms [2]. Nowadays, fraudsters can use deepfakes (synthetic media produced in the likeness of a person using deep learning technology) to do so. These technologically-enhanced activities have already caused harm. The FBI has issued warnings about malicious actors creating explicit deepfakes to harass victims [3] and to apply for sensitive jobs [4]. There have even been multiple reports of deepfakes convincing victims to part with the equivalent of hundreds of thousands of pounds [5, 6].

Barriers to accessing significant computational resources will only make it easier for adversaries to scale these activities. One report on deepfakes estimates that 90% of online content may be synthetically generated by 2026 [7]. The volume of data produced means it will be challenging for humans to vet everything without assistance. The nature of synthetic media is likely to diversify. One may primarily think of images when considering threats like deepfakes, but malicious actors will branch into other modalities like audio.

Humans are also inconsistent assessors. Studies suggest humans are overconfident in their abilities to detect falsified media [8, 9, 10]. Investigations also suggest human performance is unreliable for other safety-critical tasks. For instance, studies of X-ray baggage screeners suggest their hit rates decrease with higher workloads [11].

These results highlight the need for automated anomaly detectors to complement manual detection processes. Binary machine learning classifiers are a common but imperfect approach. Although they are highly accurate at classifying examples similar to those seen during training, their performance on out-of-distribution samples is more unreliable

[12, 13, 14]. This phenomenon means adversaries can defeat binary classifiers by slightly modifying existing threats. Collecting more threats to retrain detectors is a solution, but leads to a cat-and-mouse dynamic stalemate.

One-class classifiers aim to address the shortcomings of binary classifiers. They only use benign data to build an exemplar embedding. At inference, samples that do not resemble the exemplar features are deemed unusual. This approach is potentially more cost-efficient than binary classifiers as it does not require example threats for training. As one-class classifiers cannot overfit on anomalous training data, they should generalise better to unseen anomalies.

One-class classifiers have two core components: the underlying representation and the detector. Research on natural images suggests the former aspect is more important for performance than the latter [15, 16]. However, the questions of what a "good" representation is and how to quantify it are still unresolved.

One-class detection research also tends to focus on images. It is unclear whether these findings transfer to other modalities, which is crucial as security applications go beyond images. Therefore, this thesis aims to address the following research questions:

RQ1. What representations best facilitate anomaly detection?

RQ2. Is the choice of representation more important for performance than the choice of anomaly detector?

RQ3. What are the characteristics of good representations?

RQ4. Do similar principles for designing anomaly detectors and representations apply regardless of input modality?

In addition to revisiting one-class detectors on natural images, we analyse performance on X-ray imagery, text, speech and tabular data. Our results indicate no solution for anomaly detection that works across modalities exists. Likewise, measures of representation quality depend on the anomaly detection setup and have their caveats.

Nonetheless, our findings suggest a reasonable benign representation is low-dimensional compared to the learnt feature space. Learning representations from pre-trained neural networks trained on vast amounts of data similar to the benign and anomalous classes can enable this.

Self-supervised methods enable the extraction of suitable representations for image, text, and speech data. We demonstrate these results through empirical studies in each of these modalities. Simple self-supervised tasks (such as encouraging benign data to match a central exemplar embedding) work reasonably, raising questions on whether more complicated tasks are necessary for anomaly detection.

However, appropriate learning schemes for tabular data are unknown. Like the other modalities, we conduct detailed ablation studies on tabular data to test the effectiveness of different self-supervised approaches. Self-supervision - and broadly representation learning - is unhelpful in improving tabular anomaly detection performance.

Finally, we reconfirm that representation choice is more important than the detector, although detectors with minimal distributional assumptions are preferable.

The structure of the thesis is as follows. In Chapter 2, we provide an overview of one-class anomaly detection and representation learning. Chapters 3, 4, 5, 6, and 7 investigate anomaly detection behaviour on images, text, speech, and tabular data respectively. We summarise the findings, outline limitations, and conclude in Chapter 8.

The work in this thesis has contributed to the following publications:

1. Kimberly T. Mai, Toby Davies, and Lewis D. Griffin. Brittle features may help anomaly detection. *Women in Computer Vision Workshop at Computer at the Conference on Computer Vision and Pattern Recognition*, 2021. (Chapter 3).

2. Kimberly T. Mai, Toby Davies, and Lewis D. Griffin. Self-supervised losses for one-class textual anomaly detection. *arXiv preprint arXiv:2204:05695*, 2022. (Chapter 4).

3. Kimberly T. Mai, Sergi Bray, Toby Davies, and Lewis D. Griffin. Warning: Humans cannot reliably detect speech deepfakes. *PLoS One 18 (8), e0285333*, 2023. (Chapter 5).

4. Kimberly T. Mai, Toby Davies, and Lewis D. Griffin. Understanding the limitations of self-supervised learning for tabular anomaly detection. *Pattern Analysis and Applications*, 2024, (Chapter 7).

Beyond the thesis, we have contributed to the following:

1. Kimberly T. Mai, Toby Davies, Lewis D. Griffin, Emmanouil Benetos. Explaining the decisions of anomalous sound detectors. *The 7th Workshop on the Detection and Classification of Acoustic Scenes and Events*, 2022.

2. Lewis D. Griffin, Bennett Kleinberg, Maximilian Mozes, Kimberly T. Mai, Maria Vau, Matthew Caldwell, Augustine Mavor-Parker. Large language models respond to influence like humans. *Social Influence in Conversations Workshop at the 61st Annual Meeting of the Association for Computational Lingustics*, 2023.

3. Kimberly T. Mai, Lorenzo Pasculli, Shane D. Johnson, Lewis D. Griffin. Generative AI and homeland security: rethinking risk and response. *Under review*, 2024.

2 | Background

In this Chapter, we recap the anomaly detection landscape and the various ways to measure the properties of embeddings[1].

2.1 Anomaly detection

Anomaly detection is the task of identifying unusual instances. We label these instances as "anomalous" and the remaining instances as benign. Using the definition per Ruff et al. [1], it can be characterised as follows:

Let $\mathcal{X} \in \mathbb{R}^d$ represent the data space. We assume the benign data is 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 $\mathcal{P}$. Assuming $\mathcal{P}$ has a corresponding probability density function $p(\mathbf{x})$, the set of anomalies can be defined as follows:

$$\mathcal{A} = \{\mathbf{x} \in \mathcal{X} | p(\mathbf{x}) \leq \tau\}, \quad \tau \geq 0 \tag{2.1}$$

Where τ is a threshold. $\mathcal{P}$ 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 $\mathcal{X}$.

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.

Essentially, the decision - anomaly or benign - is a binary choice. The most straightforward

[1] This thesis differentiates between the terms "representation" and "embedding". Representation refers to the learnt space, whereas embeddings are the samples that map into the representation space.

way of facilitating this is with a supervised binary classifier. However, supervised models come with disadvantages. They require labelled anomalies for training. Collecting such samples can be costly and challenging as anomalies rarely occur. Moreover, supervised classifiers require re-training to categorise unseen anomalies correctly.

In light of the shortcomings of supervised classifiers, we focus on one-class models. One-class models only use benign data for training. We can train a model directly on the data, or pre-process the data to extract relevant features. Traditional feature engineering pipelines were hand-crafted. Nowadays, one can use neural networks.

The model compares the embedding of a test datum to the exemplar training embedding at inference time. The more dissimilar the datum is, the more likely it is an anomaly. The degree of acceptable dissimilarity depends on a set threshold. This threshold is a trade-off between false accepts and rejects.

Appropriate thresholds are task-dependent. For example, it may be more acceptable to have higher false reject rates when inspecting baggage at the airport compared to facial verification systems to unlock personal smartphones. We use the area under the receiver operating curve (AUROC) as the primary evaluation metric to avoid tuning different thresholds for each application mentioned in the thesis and for consistency with other work. The receiver operator curve is a graphical depiction of the true positive against the false positive rate at various thresholds. We can consider AUROC as the probability that a randomly selected anomaly will be ranked as more abnormal than a benign sample. Scores fall between 0% and 100%. A score of 50% indicates a detector cannot distinguish between anomalies and benign data points, while a score of 100% signals perfect anomaly discrimination.

Although AUROC tends to be more stable for imbalanced datasets than accuracy, it may not fully reflect model performance in situations of extreme imbalance. AUROC may fail to highlight incorrect classifications of rare anomalies. As a result, a high AUROC score in an imbalanced dataset could signal that a model only learns to correctly identify benign samples.

Anomalies can be of different types. Two important categories are [17]:

1. **Semantic**: The instance is an unusual object category (e.g., firearms in baggage).

2. **Fine-grained**: The instance is a typical object category. Instead, the instance contains unexpected patterns in specific segments [18], such as unusual textures in baggage due to power explosives [19].

This thesis focuses on semantic anomalies. Future work could extend our investigations to fine-grained anomalies.

Anomalies are further categorised by how far they deviate from benign data. The distribution of near out-of-distribution (OOD) samples resembles the benign distribution more closely than far OOD points [20]. As a concrete example, if the training data contained X-ray images of airport baggage, then X-ray images of baggage with firearms would be near OOD. In contrast, camera photos of luggage containing firearms would be far OOD as they are from a different domain. We primarily focus on near OOD samples by partitioning existing classification datasets to construct anomalies (for example, we treat the "automobile" class in CIFAR-10 [21] as anomalies and the remaining nine classes as benign).

Benign data also fall into two main categories: unimodal and multimodal. The data points in the unimodal setting are more similar to each other than the multimodal instance. Using CIFAR-10 as an example, an unimodal configuration might only include "cat" as the benign class, whereas a multimodal configuration might incorporate all animals (i.e., cat, deer, dog, frog, horse). We use both configurations.

2.1.1 Related fields

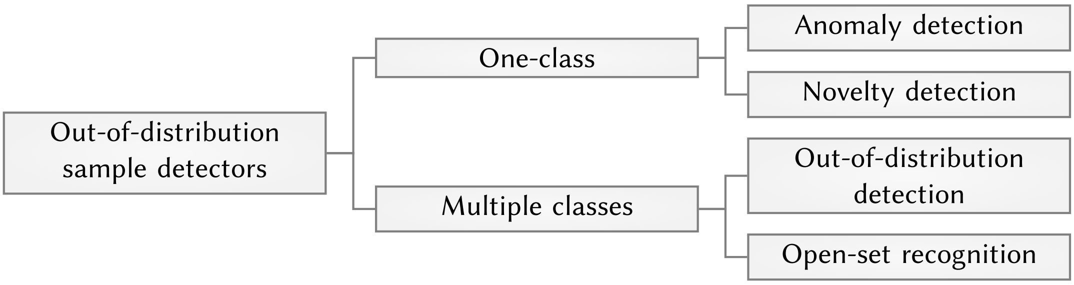


Figure 2.1: Taxonomy of out-of-distribution sample detectors.

Anomaly detection closely aligns with novelty detection, open-set recognition and out-of-distribution detection [22]. Although all fields involve detecting OOD samples, the literature uses out-of-distribution detection to refer to models explicitly designed to distinguish between multiple classes.

All fields assume only benign data are available at training. However, anomaly and novelty detection group training data into one class. In contrast, semantic labels are available for the benign subclasses in open-set recognition and out-of-distribution detection.

Anomaly and novelty detection differ in their treatment of OOD samples. Anomaly detection treats them as abnormal, whereas novelty detection views them as merely unseen. Consequently, future iterations of novelty detectors may incorporate previously identified unseen data points into training.

Open-set recognition and out-of-distribution detectors assess test data in different ways. Open-set models simultaneously classify known classes of in-distribution data while flagging unseen classes. In contrast, out-of-distribution detection solely focuses on identifying unseen samples. They use the subclasses to aid decision-making.

Methods designed for one field are adaptable to another. One can use metadata or augmentations to create new subclasses to turn a one-class configuration into a multiple-class one [23]. Conversely, one can combine the subclasses into one group.

We provide further details on OOD sample detectors in Appendix A.

2.2 Representation learning

Representation learning is a machine learning paradigm that aims to transform a raw data space into one with concise and meaningful features. Previous one-class anomaly detection methods that used representation learning relied on traditional feature engineering, such as Mel spectrograms for audio [24] and histograms of gradients for images [25]. However, using features from neural networks is now the default choice and has improved performance significantly [26].

Features from pre-trained classification networks transfer well to other tasks, even when the downstream modality is different [27]. For instance, Palanisamy et al. [28] showed that convolutional neural networks (CNNs) trained on ImageNet [26] are a strong baseline for audio classification. Fine-tuning the neural network with the target dataset can boost performance. However, supervised learning may not be feasible because labels might not be available. For instance, the exact contents of benign luggage seen in X-ray scanners may be too complicated to annotate. The range of items varies widely, depending on a passenger's travelling purposes. Therefore, using the OOD confidence of a model classifying different items inside luggage may be infeasible.

Self-supervised learning overcomes this issue by using intrinsic properties of the data as labels. They use pretext tasks to generate labels from unlabelled data [29]. Examples of pretext tasks include colourising greyscale images [30] or predicting the next word in a sentence [31, 32]. Understanding the typical characteristics of a domain allows one to create an appropriate pretext task. For instance, colourisation requires knowledge of object boundaries and semantics. These aspects are useful for image classification [33, 34].

2.2.1 Self-supervised pretext tasks

Some categories of pretext tasks are as follows. Balastriero et al. [29] covers these in more detail.

Classifying perturbations: Each training datum is subject to a perturbation randomly selected from a fixed set, such as rotating the input data [35] or reordering patches in an image [36]. 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 [31] or filling in masked areas of an image [37, 38].

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 [39, 40]. InfoNCE takes augmented views of the same data point as positives and learns to group them while pushing away other data points. 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 [41] 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 embeddings. This regularisation ensures the neural mapping covers complementary information across the representation space.

2.2.2 Measuring embeddings

Different pretext tasks learn different features. Variances across tasks can make it unclear what features are more desirable than others. Understanding the importance of features would help us understand how neural networks work and how to improve them. For example, it would help us recognise whether the model would benefit from data augmentation or architectural changes.

Neural networks are less interpretable than shallow machine learning models like linear regression, partly due to the large number of parameters and nonlinearities [42]. The field of adversarial machine learning highlights the difficulty of interpreting neural networks. For example, adding imperceptibly small amounts of noise can cause an image classifier to label a panda as a gibbon [43].

The one-class anomaly detection setup adds another layer of complexity. Embedding quality is only measurable with benign data. Yet, a good representation space needs to generalise poorly to out-of-distribution data. Hence, measures that rely on the training data might be misleading.

There are still ways to measure embedding quality in the one-class setting. These methods fall into two categories: explainability (analysing task outputs) or interpretability (analysing the internal properties of embeddings) [44]. Analysing task outputs involves studying how the extracted embeddings perform on particular tasks. These embeddings are not necessarily from the final layer of neural networks. Some representation learning studies find that using intermediate embeddings can be better for anomaly detection because the later layers of neural networks specialise in the training task, while earlier layers learn more generalised features [40, 45, 46]. In contrast, measures that analyse internal properties look at the properties of data after they have been transformed by neural networks. We summarise some examples relevant to one-class anomaly detection below.

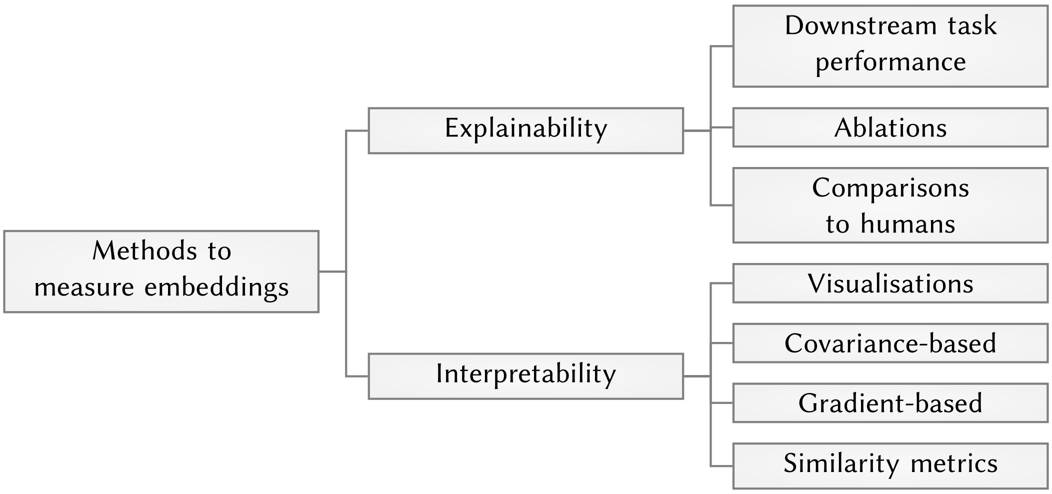


Figure 2.2: Diagram showing methods for assessing neural network embeddings.

Downstream task performance. We freeze the neural network of interest and extract relevant features. We use these features to train a detector and evaluate a downstream task, like anomaly detection or classification. We use performance metrics like accuracy to implicitly measure embedding quality. This setting relies on having a validation set that faithfully reflects the test conditions. For example, if the trained model was a spam detector, the spam emails in the validation set should contain content similar to spam in real-life settings.

Ablations observe how outputs change following input or architecture modifications. For example, the work of Geirhos et al. [33] showed that ImageNet-trained CNNs were biased towards texture by training ResNets to classify the ImageNet categories and varying the input features (natural colour, greyscale, silhouettes, canny images and textures). Performing ablations allows us to understand if trained models have biases or limitations. However, extensive ablation studies involve consistently training new models, which is expensive. Additionally, using ablations for hyperparameter tuning raises the prospect of overfitting. This overfitting can be problematic if there is a misalignment between the ablations and the ultimate purpose of the model.

Human comparisons. Models that make the same classifications or mistakes as humans indicate they use similar characteristics for decision-making. For example, RotNet, which encourages a model to predict the correct orientation of input images, relies on the regularity that many semantic classes have a non-uniform distribution of orientations [35]. One instance is cars, where wheels are generally in contact with the ground.

Human comparisons can be as primitive as measuring how well models perform on an annotated dataset, such as classifying ImageNet. More sophisticated studies involve recruiting participants and comparing their responses to the model's outputs. The texture study by Geirhos et al. recruited 97 participants to classify the same input features used to train the ResNet models [33]. Unlike the ResNet models, human participants could still categorise images with altered textures. From these results, Geirhos et al. concluded that ResNet models relied more on texture to make classification decisions.

However, human studies come with considerations. Human participants can be costly to recruit. Their decisions are also often error-prone and biased. One study suggests ImageNet and CIFAR-100, two widely used datasets in computer vision research, contain annotation errors close to 6% in their test sets [47]. Another study on out-of-distribution detection suggests a 50% overlap between in-distribution ImageNet-1K and commonly-used out-of-distribution test sets [48]. These results imply the performance of classification or out-of-distribution detectors may not faithfully indicate if one model is indeed better than another. Human responses can also encode biases. One open-source large-scale dataset, 80 Million Tiny Images [49], was withdrawn after a study found the annotations contained derogatory terminology. As a result, models trained on this dataset exhibited offensive biases [50].

Visualisations are useful in illustrating embedding behaviour. They aim to present embeddings in a way humans can understand. Dimensionality reduction techniques like PCA and t-SNE aim to capture the most crucial data directions [51]. For example, Gao et al. visualised word embeddings from a vanilla transformer using singular value decomposition and found they occupied a narrow space [52]. Concurrent and subsequent works confirmed this phenomenon [53, 54, 55, 56].

However, dimensionality reduction techniques have caveats. They use assumptions that might not apply. PCA is a linear method that disregards the nonlinearities encoded in deep neural networks.

T-SNE assumes the data follows a t-distribution in the lower dimensional space and is better at modelling local relationships. It is also not deterministic. Re-running the algorithm generates different results. In addition, t-SNE cannot transform unseen data and has to be run on test data. Therefore, the visualisations can be misleading in a one-class setup.

Covariance-based methods typically summarise the eigendecomposition of the covariance matrix to a scalar value. Other works have used these scalar values to inform model and hyperparameter selection.

The first measure mentioned in this review, α-ReQ, takes inspiration from neuroscience. A 2019 study by Stringer et al. analysed the encoding dimensionality of natural images in the visual cortex of mice [57]. The encodings were high dimensional, and correlations obeyed an $\frac{\alpha}{n}$ power law, where $\alpha \approx 1$. They computed the correlations by plotting the eigenvalue spectrum of the covariance matrix of the encodings on a log-log scale and measured the gradient. This gradient is termed α. The correlations were not due to the power spectra of natural images [58], as this behaviour persisted after whitening the stimuli presented to the mice. An α less than 1 indicates a dense encoding, whereas an α larger than 1 indicates sparser encodings. Stringer et al. argue that $\alpha \approx 1$ is desirable. Slower eigenspectrum decays would increase sensitivity to small changes in input stimuli while faster decays would decrease sensitivity to changes in input stimuli.

α-ReQ used the α approach to measure the quality of learnt embeddings in self-supervised image classification models [59]. They found an α closer to 1 was a "sweet spot" for downstream classification performance. However, they emphasise this value is necessary but insufficient for good downstream performance. Self-supervised models using CNNs as backbones exhibited positive correlations between classification accuracy and α, whereas transformer-based models showed negative correlations. In addition, the authors found that correlations between α and performance were weaker in the earlier layers of the models and hypothesised this is due to earlier layers learning more broad task-invariant features.

RankMe evaluates embedding quality by estimating the rank of the embeddings [60]. They hypothesise larger values are better because they exhibit less dimensional collapse. The score is as follows:

$$\text{RankMe}(Z) = \exp\left(-\sum_{k=1}^{\min(N,K)} p_k \log p_k\right) \tag{2.2}$$

$$\text{with} \quad p_k = \frac{\sigma_k(Z)}{\|\sigma(Z)\|_1} + \epsilon, \tag{2.3}$$

Where Z are the training embeddings, σ_k is the k-th singular value of Z and ϵ is a stability constant. The study used RankMe to perform self-supervised model selection and showed the metric correlated with classification accuracy. The authors demonstrated that α-ReQ and RankMe exhibited similar performance when used as a model selection metric but found that RankMe outperformed α-ReQ in cases of dimensional collapse. However, RankMe cannot be used to compare architectures. The authors emphasise it is only usable when comparing identical architectures.

Another similar metric is the area under cumulative explained variance (AUCEV). Li et al. [61] used AUCEV to establish why non-contrastive self-supervision approaches (like BYOL [62] or SimSiam [63] that use stop-gradients instead of contrastive losses to learn embeddings) perform well even though the global minimum is a collapsed embedding. They found that model capacity closely aligns with classification performance. If the model is too small compared to the dataset, it leads to dimensional collapse. They use AUCEV to measure the extent, where there are K singular values in total:

$$\text{AUCEV} = \frac{\frac{1}{K}\sum_{i=1}^{K}\sum_{j=1}^{i}\sigma_j}{\sum_{k=1}^{d}\sigma_k} \tag{2.4}$$

Values closer to 50% indicate no collapse, whereas 100% indicates severe dimensional failure. Hence, AUCEVs closer to 50% are more desirable. Unlike α-ReQ and RankMe, Li et al.'s experiments only cover non-contrastive self-supervised models and use ResNets as backbones.

Gradient-based methods rely on the backpropagation aspect of deep neural networks. They examine gradient magnitudes given some input data. A sizeable gradient update suggests the model is learning a new concept. Therefore, if a model trained on a particular semantic class encounters a datum seen previously, the gradient should be smaller. Consequently, gradient magnitudes are proxies for measuring the tightness of an embedding. Simon-Gabriel et al. show the ℓ_p norm of gradients calculated at the loss $\mathcal{L}$ corresponds to vulnerability to adversarial perturbations [64]. Higher norms correspond to higher vulnerability. This method cannot be applied to non-neural anomaly detection methods because it requires a model that can backpropagate.

Similarity metrics correspond to the assumption in anomaly detection that benign embeddings should be compact [65, 66, 67]. One way to measure this is by calculating the cosine similarities across the training samples [68]. Another variation involves calculating the average cosine similarity against a mean embedding [66, 67].

Similarity metrics can also compare embeddings from different models or layers. Alternative methods like centred kernel alignment (CKA) [69] commonly measure cross-model similarities. Unlike cosine similarity, CKA is invariant to linear transformations. Therefore CKA can compare different representation spaces. However, CKA is more complicated to run than cosine similarity. The first step involves computing kernel matrices for the two embeddings. If $\mathbf{X}^{m_1}$ and $\mathbf{X}^{m_2}$ are two embeddings with dimensionalities m_1 and m_2, their similarity matrices after the kernel step are as follows:

$$\mathbf{K}^{m_1} = k(\mathbf{X}^{m_1}, \mathbf{X}^{m_1}), \quad \mathbf{K}^{m_2} = k(\mathbf{X}^{m_2}, \mathbf{X}^{m_2}) \tag{2.5}$$

The matrices are zero-centred to $\mathbf{HK}^{m_1}$ and $\mathbf{HK}^{m_2}$, where $\mathbf{HK}$ is the centred matrix. The two matrices undergo comparison using the CKA score. The CKA score typically uses the Hilbert-Schmidt independence criterion, which measures the independence between two distributions and is normalised. CKA scores range between 0 and 1. Scores closer to 1 suggest higher alignment.

$$\mathrm{HSIC}(\mathbf{X}^{m_1}, \mathbf{X}^{m_2}) = \frac{1}{(n-1)^2}\mathrm{tr}(\mathbf{HK}^{m_1}\mathbf{HK}^{m_2}) \tag{2.6}$$

$$\mathrm{CKA}(\mathbf{X}^{m_1}, \mathbf{X}^{m_2}) = \mathrm{HSIC}(\mathbf{X}^{m_1}, \mathbf{X}^{m_2}) \;/\; \sqrt{\mathrm{HSIC}(\mathbf{X}^{m_1}, \mathbf{X}^{m_1})\mathrm{HSIC}(\mathbf{X}^{m_2}, \mathbf{X}^{m_2})} \tag{2.7}$$

The kernel operations mean CKA is more expensive to compute than cosine similarity and is less interpretable. The population characteristics of the input data can also confound CKA and lead to misleading similarities. Cui et al. propose a fix by regressing out the input features [70].

2.3 Landscape of representation learning for anomaly detection

Representation learning approaches for anomaly detection use deep neural networks end-to-end or as feature encoders. Both rely on the same assumption: the embeddings learnt by the neural networks can characterise benign data but not anomalies.

End-to-end approaches work on the principle that neural networks behave differently on benign data and anomalies. When dealing with anomalies, losses are higher [46, 71, 72], predictions from discriminative models should be more uncertain [13], or log-likelihoods in generative models are lower [73]. However, this assumption does not always hold in practice.

Nalisnick et al. [73] trained generative models to reconstruct images from multiple datasets (MNIST, FMNIST, SVHN, CIFAR-10, CIFAR-100). They used the log-likelihood to score samples, anticipating that OOD samples should have lower values. The log-likelihood scores for the OOD datasets were often *higher* than the training datasets across several generative models. Possible reasons include poor calibration between the actual underlying distribution and the fitted distribution, the bias of deep generative models towards low-level statistics, and the curse of dimensionality [74, 75, 76]. Alternatively, the OOD datasets could have more straightforward characteristics than the training set. The generative models might contain the features to generate these out-of-distribution samples. For example, Zenati et al. [77] combined the losses from the generator and discriminator of a generative adversarial network (GAN) to score anomalies. When training their GAN on MNIST, the model treated digits of the anomalous class "1" as more typical than the training set classes ("0", "2",...,"9").

Feature extraction approaches feed neural network outputs into a shallow anomaly detector. k-nearest neighbours (k-NNs) are a popular choice due to their nonparametric nature [15, 16, 67, 78, 79], but other approaches like one-class support vector machines (OCSVMs) and Gaussian mixture models (GMMs) have also been used [80].

Embeddings do not always originate from the final layer. Intermediate embeddings have also demonstrated good performance [80]. For instance, Andrews et al. [45] extract intermediate embeddings from a sparse autoencoder to train OCSVMs. They found these embeddings worked better on average than using input features or the residuals from the autoencoder. However, there is no method for identifying which intermediate embeddings work best. Intermediate embeddings are also not guaranteed to work better. For instance, Xu et al. [81] trained OCSVMs with intermediate embeddings of BERT [82] and RoBERTa [83] to detect OOD samples on two datasets. They found the penultimate layers worked best on one dataset but there was no best-performing layer for the second dataset.

There is no consensus on whether end-to-end methods are better than feature extraction approaches. Using models end-to-end removes the need to assemble and tune different components in the anomaly detection pipeline (for example, selecting the best-performing shallow detector). In contrast, outputs from shallow detectors are more straightforward to interpret and diagnose.

Benchmarking studies that compare different OOD detectors show k-NN can outperform end-to-end models on near OOD but not far OOD instances when using a ResNet-50 pre-trained on ImageNet as a backbone [84]. However, these benchmarking studies concentrated on the scenario where multiple labels are available for the training data, which differs from the one-class focus in this thesis.

More broadly, representation learning methods for anomaly detection focus on computer vision and use ImageNet as the benign distribution. Follow-up studies would benefit from applying representation learning methods to more diverse scenarios.

3 | Images

Reviewing images is a common way of assessing safety. Examples include identifying bogus biometric attempts [85], flagging unusual vehicle movements [86], and pinpointing suspicious items in parcels [17]. The contents of images can vary greatly and it can be challenging for humans to keep up with the changing nature of anomalies. Therefore, complex and varied images emphasise the need for automated detectors.

One might use supervised classifiers as an initial option. However, it is challenging to collect a sufficient number of anomalies for training. For instance, a training set for a supervised firearms classifier suffers from this problem. Finding firearms in the wild is rare. The difficulty of collecting training anomalies motivates the use of one-class anomaly detectors. Regardless, there might not be sufficient data to train a one-class model from scratch. For instance, X-ray images of luggage have vastly different characteristics from natural images and require specialist equipment to collate. One could adapt features from models specialising in processing natural images for a domain like X-rays. Pre-trained models trained with natural images are widely available. However, it is unclear whether using them to evaluate X-ray images would result in a domain shift that affects detectability.

We compare how different representation spaces affect anomaly detection performance. To do so, we use knowledge distillation [87] to construct a setting where the model should be able to represent benign data but not anomalies. We select a teacher network trained on a pretext task. This task originates from a more complex pre-trained classification task or self-supervised learning. A student network that only sees inlier training data learns to match the internal embeddings of a frozen teacher network. Using the idea that regression models extrapolate poorly to unseen data, we expect the embeddings of the student and teacher to differ more on anomalous images compared to benign images [88]. We use mean squared error to score anomalies, as the extent of disagreement between student and teacher should show through higher regression errors. Although there have been works that have used this failure-to-extrapolate idea [88, 89, 90], these works have fixed the teacher representation spaces and have not explored how varying the teacher can affect detection.

We derive representation spaces from various sources, including an X-ray security dataset containing staged threats [17]. Our approach outperforms the previous best anomaly detection scores for that dataset, boosting the AUROC score from 92.65% to 96.41%.

In addition, our results suggest embeddings with features susceptible to adversarial perturbations may be better candidates for anomaly detection. To summarise, our contributions are as follows:

1. We conduct an empirical study to compare the suitability of multiple candidate embeddings for anomaly detection.
2. We confirm the choice of representation space is more important than the anomaly detection method.
3. We demonstrate that separability between anomalies and benign data does not ensure reasonable detection performance.
4. We show that embeddings more prone to adversarial perturbations may be more suitable for anomaly detection. We also propose a score to measure the vulnerability.

This work was presented at the Women in Computer Vision Workshop at CVPR [91].

We outline the anomaly detection literature in §3.1 and describe our approach in §3.2. We analyse our results in §3.3. Finally, we summarise and conclude in §3.4.

3.1 Background

Earlier approaches for image-based anomaly detection used hand-crafted feature engineering (such as colour histograms [92] or textural features [93]) to pre-process high-dimensional images.

The advent of deep learning enabled the learning of features that specialised more to the training distribution [43]. Generative models, particularly autoencoders, are popular end-to-end approaches that simultaneously learn relevant features and assess anomalousness [45, 71, 72]. However, evidence showed that these models often deem OOD data less anomalous than benign data [73]. One explanation speculates that generative models fail to model the actual distribution sufficiently [74].

Instead of using the sampled training data to learn representation spaces from scratch, later works use transfer learning [94]. Representations formed from models pre-trained on mass amounts of data should have fewer discrepancies between the modelled and actual distributions. The diversity of large-scale natural image datasets like ImageNet [95] makes transfer learning a sensible starting point for image-based anomaly detection. Transfer learning methods use the pre-trained models directly or fine-tune with the benign data before feature extraction. Anomaly detection methods feed the embeddings through shallow detectors [94, 96] or apply feature engineering (such as binarisation [17]) to enhance the difference between benign and anomalous inputs.

The size of pre-trained models means it can be expensive to extract features directly. Knowledge distillation is a technique closely linked to transfer learning that addresses this issue [87]. This technique involves two models: a teacher and a student. The student model tends to be more computationally efficient while retaining salient features of the teacher. Instead of training the student from scratch, the student learns to mimic the outputs of the teacher. These outputs can be the teacher's predictions or intermediate embeddings.

Knowledge distillation has improved both classification and anomaly detection [89, 97]. Works hypothesise knowledge distillation is helpful because the teacher logits provide more information about how classes relate to each other than discrete labels [87, 97].

Anomaly detectors that use knowledge distillation assume the student network represents benign data similarly to the teacher but diverges on anomalies. Bergmann et al. propose an ensemble-based method for fine-grained anomaly detection [89]. An ensemble of students learns to mimic the pre-trained teacher output by minimising the patch-based distance between the mean student embedding and the teacher embedding. The anomaly score combines the reconstruction loss between each student and teacher with the average amount of disagreement between the students. As this method is for fine-grained anomaly detection, it relies on combining pixel patch embeddings, which is expensive.

Neural networks tend to be susceptible to adversarial perturbations. An imperceptible change to the input data can alter neural network outputs. Studies suggest vulnerability is related to the magnitude of the loss gradient [64, 98]. The occurrence of adversarial examples suggests neural networks rely on brittle features for learning representations [43]. Works argue that these brittle features are not at odds with downstream performance. Instead, they claim brittle features are a direct product of a model's sensitivity to well-generalising features in data [99, 100].

3.2 Method

Although several works use pre-trained embeddings for anomaly detection, embeddings from ImageNet-based classifiers are the most common choice [13, 17, 96, 89]. These embeddings may not be suitable when there are domain shifts, for instance, from natural images to X-rays. Therefore, our research questions are as follows:

1. What representations and embeddings are best for image-based anomaly detection?
2. What are the properties of the most suitable embeddings?

Our anomaly detection setup relies on a reconstruction principle. A good embedding should contain features sufficient to reconstruct benign instances but insufficient to reconstruct anomalies. Our approach is similar to using autoencoders and a reconstruction loss as the anomaly score [71] but fixes a flaw with autoencoders. Some works show that vanilla autoencoders can construct more rudimentary OOD classes [73, 77]. For example, Zenati et al. [77] showed that autoencoders can reconstruct the anomalous class "1" in MNIST if the remaining numbers are benign.

Ren et al. [74] suggest this behaviour happens because the reconstruction relies on shortcut features. For example, in MNIST, the proportion of zeros (the number of pixels belonging to the background in an image) is a confounding factor and affects reconstruction. They term this proportion of zeros a "background" representation. Ren et al. propose fixing this issue by introducing likelihood ratios. The likelihood ratio statistic compares a representation that only models benign data against a background representation that can model other semantic classes beyond the benign class.

Our teacher-student approach uses a similar principle. We train the teacher on more complex pretext tasks, sometimes on more diverse data. The student only uses benign data specific to anomaly detection. Its task is also simple. It learns to match the teacher's outputs. As a result, the student learns a representation space more specific to the benign data, whereas the teacher learns a representation that captures background information. We emulate the likelihood ratio scenario by passing a test datum through the teacher and student and comparing embeddings. The student and teacher should output similar embeddings for benign data and divergent embeddings for anomalies. We visualise our approach in Figure 3.1.

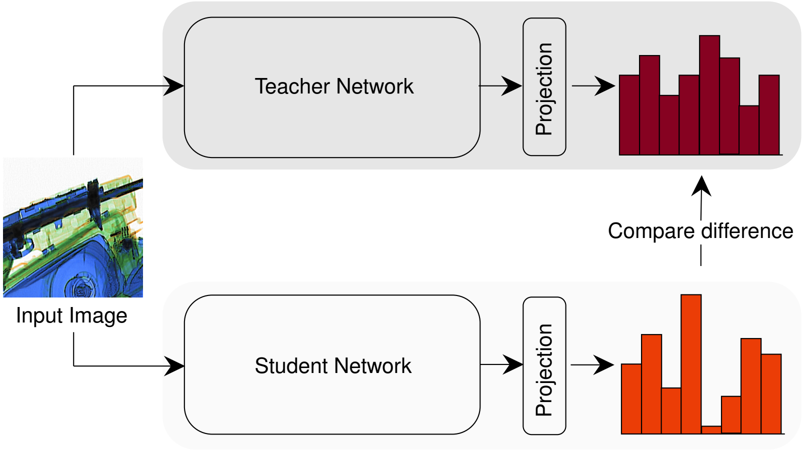


Figure 3.1: Schematic of the knowledge distillation architecture.

First, we train a teacher network on a pretext task. We outline all pretext tasks in §3.2.1. In all cases, the teacher does not encounter any anomalies during the training stage. We then freeze the teacher weights and randomly initialise a student network. We feed the same input image to the student and teacher. The student learns to match their output with the teacher. We optimise the student using mean squared error (MSE). We experimented with ensembling and other loss functions but did not notice significant improvements.

Our inference pipeline is similar. We pass a test image through both networks and use the MSE as the anomaly score. For an input datum x, let the output of the student be $f(x)$ and the output of the teacher be $h(x)$. The anomaly score $A(x)$ is then defined as:

$$A(x) = ||h(x) - f(x)||^2 \tag{3.1}$$

3.2.1 Pretext tasks

We use generative and discriminative tasks to cover the range of learning objectives. The tasks are as follows:

Classification from other datasets. We train neural networks to classify a dataset separate from the one we use for anomaly detection. We use a standard classification training regime: we use the provided annotation labels and train with cross-entropy loss.

RotNet uses a perturbation-type self-supervised task [35]. Each input image rotates by a multiple of 90 degrees $\{0^{\circ}, 90^{\circ}, 180^{\circ}, 270^{\circ}\}$. The model learns to predict the rotation using the angle as the label. RotNet learns orientations and subsequent features that are important for identifying semantic classes. The networks use cross-entropy loss.

SimCLR is a self-supervised clustering task for images [40]. It trains a Siamese neural network to distinguish between two augmented views of the same image. The training process uses a contrastive loss function called NT-Xent. NT-Xent compares images in a minibatch. Two differently augmented views of the same image are the positive pair, while the remaining samples in the minibatch are negative. This loss encourages the network to learn the semantic features in images and makes the network invariant to factors irrelevant to semantic classification, such as sharpness and colour. In the original implementation, the authors experimented with different augmentations. They found random cropping, flipping, colour distortion and Gaussian blur worked best. We use the same augmentations.

Autoencoders (AE) are unsupervised networks that learn to reconstruct the input data [101]. An autoencoder typically includes a bottleneck in its architecture. It projects the input data through this bottleneck so that it is encouraged to learn more efficient embeddings of the data and to ignore irrelevant noise. We optimise the autoencoders using MSE.

Denoising autoencoders (DAE) are extensions of AEs. They learn to reconstruct the data using a corrupted version as input [102]. By learning to remove the noise in addition to recreating the data, DAEs should be more robust to noise and irrelevant variations in input data. However, adding noise makes the training process more computationally expensive than AEs. We optimise the DAEs with MSE and apply Gaussian noise ($\sigma = 1$) to the input data.

We also include two control representations:

Random weights (Random). We randomly initialise the teacher weights. We apply the default Glorot initialisation [103] because we use PyTorch to implement the models. Glorot initialisation aims to preserve the variance of the input signal as it passes through a network to prevent vanishing gradients by ensuring zero means and maintaining the value of the variance of the input in every layer. Random weights provide a floor for performance.

Supervised anomaly detection (Supervised) uses a teacher network that classifies benign and anomalous training samples. Although this is not a realistic representation for one-class anomaly detection, this setup provides a ceiling for the best teacher representation. The network specifically learns to categorise typical samples and anomalies from the same distribution as the test distribution.

3.2.2 Comparisons

We compare our knowledge distillation framework to shallow anomaly detectors trained directly on embeddings extracted from the teacher. We use AUROC to benchmark all anomaly detection methods.

Mean squared error. We pass all the training data through the teacher and extract the newly transformed embeddings. We calculate the mean embedding across the training samples. We use the MSE between a test datum (transformed by the teacher) and the mean embedding as the anomaly score. This scoring method uses the same assumption as the centre loss setup [66]: benign data should be close to the prototypical (teacher) embedding.

Mahalanobis distance. In addition to the mean embedding, we calculate the covariance of the training data from the newly transformed embeddings. We calculate the full and diagonalised covariance to understand whether including correlations between variables improves performance. We use the Mahalanobis distance between a test datum and these values to calculate anomaly scores.

We also compare the performance of the anomaly detectors to the supervised classification performance. We freeze the teacher networks, append a linear head and fine-tune the networks to classify between benign samples and anomalies. CIFAR-10 is the exception: we fine-tune the network to categorise all ten classes.

3.2.3 Datasets

We evaluate anomaly detection performance on the following datasets:

Cats vs. Dogs (CvD) contains 25,000 coloured images of cats and dogs split equally between the two classes [104]. For each class, we allocated 10,000 images for the training set and the remaining 2,500 images into the test set. As image sizes vary, we resize each image to 32 × 32 pixels. The anomaly detectors only used one class for training. The images of the other class are deemed anomalous.

CIFAR-10 contains 60,000 32 × 32 colour images split equally across ten classes (*airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck*) [21]. In total, there are 50,000 training images and 10,000 test images. There are no overlaps between the classes. We evaluate unimodal and multimodal configurations for all ten classes, resulting in twenty configurations overall. For example, for the unimodal configuration, we could pick *airplane* as the benign class and set the remaining classes as anomalies.

Plant Pathology 2020 is a Kaggle dataset. It contains 3,651 colour images of apple leaves taken under different light, angle, and noise conditions [105]. Of those, 1,821 images were labelled and categorised into one of four categories for the Kaggle competition: healthy (516), containing apple scabs (592), containing cedar apple rust (622), or containing multiple diseases (91). We use 80% of the labelled healthy images for training and the remaining 20% for testing. We sampled an equal number from the other classes to serve as anomalies during testing. We do not use the unlabelled images. The original images are 2048 × 1065 pixels. We resize each image to 224 × 224 to fit computational constraints.

The **X-ray** dataset consists of 5,000 stream-of-commerce (SoC) and 234 staged threat (threat) parcels collected from a UK parcel distribution centre [17]. The SoC parcels consist of benign objects, whereas the threat parcels include a firearm. All parcels have dual views, which show the same contents at perpendicular angles. They are false-coloured with dual-energy imaging. The images are 764 pixels high, while the width varies. SoC parcels have a median width of 676 pixels, whereas threat parcels have a median width of 990 pixels. We show examples from the dataset in Figure 3.2.

We use the same pre-processing steps as Griffin et al. [17], who initially introduced the dataset (Figure 3.3). We cropped the images to remove extra air and split them into 224×224 patches using a stride of 112 or less so that both views were fully covered. We only use SoC patches for training. The models only encounter the threat patches during the evaluation stage. As we fed patches into the networks instead of full images, we used the maximum

MSE across an image's patches as the anomaly score.

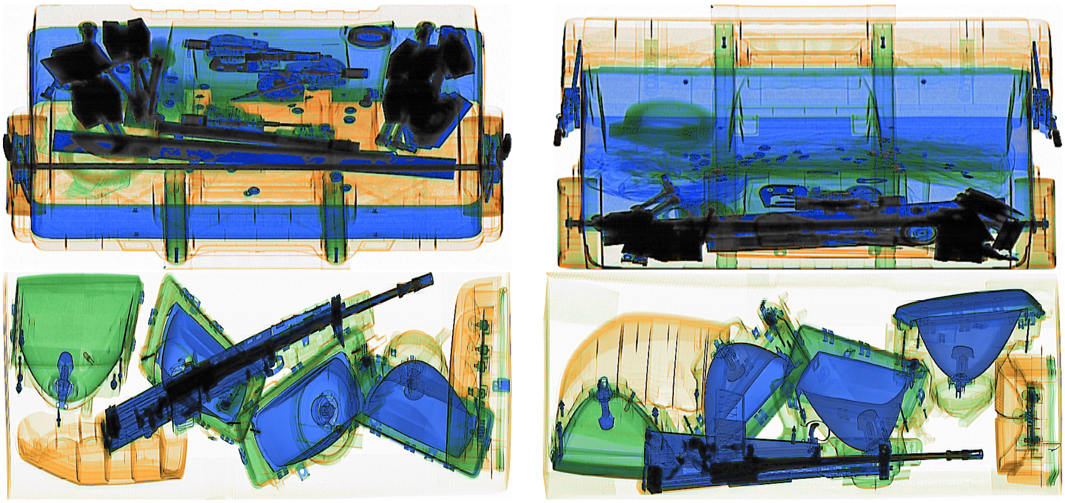


Figure 3.2: Example dual-view images from the X-ray dataset. The top row is a SoC example and the bottom is a staged threat example.

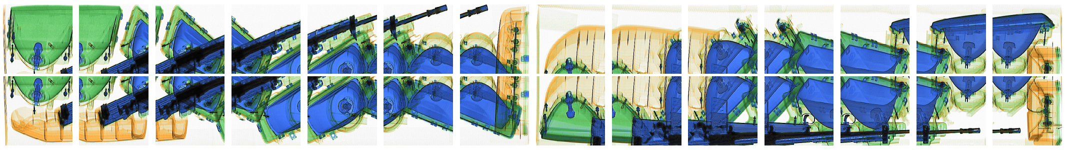


Figure 3.3: Example of how the X-ray images are pre-processed using the threat example from 3.2. SoC and threat images undergo the same pre-processing steps.

3.2.3.1 Additional datasets

We train teacher networks with additional datasets. We use these teacher networks to investigate how transfer learning from other representations affects detection performance.

STL-10 is an image recognition dataset inspired by CIFAR-10 [106]. There are ten classes (*airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck*). Each class contains the same number of coloured images: 500 training images and 800 testing images. All images are 96× 96 pixels. The dataset also includes 100,000 unlabelled images, which are for unsupervised learning. We resize each image to 32 × 32 pixels to train the teacher models. We only use labelled data for the auxiliary classification tasks, while we also include unlabelled data to train the RotNets and AEs.

Fashion MNIST (FMNIST) is a greyscale dataset containing images from ten clothing classes (*t-shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, ankle boot*) [107]. There are 60,000 training and 10,000 testing images, each of size 28 × 28. The classes are equally balanced. We resize each image to 32 × 32 for the auxiliary classification task.

Table 3.1: Pretext tasks and datasets used to pre-train the teacher and generate fixed representations for each anomaly detection dataset, excluding randomly initialised weights which are used for all datasets. We italicise the supervised (ceiling) representations. The datasets in the table cells were used as training data for the pretext task.

	Pretext Task				
Evaluation dataset	Classification	RotNet	Autoencoder	Denoising Autoencoder	SimCLR
Cats vs. Dogs	STL CIFAR-10 *Cats vs. Dogs*	STL CIFAR-10 Cats vs. Dogs	STL CIFAR-10 Cats vs. Dogs	STL CIFAR-10 Cats vs. Dogs	Not used
CIFAR-10	STL FMNIST *CIFAR-10*	STL FMNIST CIFAR-10	STL FMNIST CIFAR-10	STL FMNIST CIFAR-10	CIFAR-10
Plant Pathology	Plant Village ImageNet *Plant Pathology*	Plant Pathology	Not used	Not used	Not used
X-ray	ImageNet *X-ray*	X-ray	Not used	Not used	Not used

Plant Village contains 54,309 healthy and unhealthy leaf images spanning fourteen crop species [108]. The species further divide into healthy or diseased categories, leading to 39 disjoint classes. We train the teacher model to classify between healthy and unhealthy plant images.

ImageNet is a large-scale image recognition dataset [95]. It has more than 14 million images. These images have labels corresponding to a lexical database called WordNet [109]. We do not train the teacher models on ImageNet ourselves. Instead, we initialise the teacher with pre-trained ImageNet weights.

We summarise the pretext tasks and datasets in Table 3.1.

3.2.4 Architectures

We fix the architecture for each dataset. The students and teachers have identical architectures for all experiments. This choice is not a requirement for the method; we choose to do so to minimise architecture search. We used the Adam optimiser [110] to train all students with a learning rate of $1e-5$ for 20 epochs, as we found higher learning rates led to training instabilities. We did not apply augmentations to the images when training the students, as we found they could erode features that distinguish benign images from anomalies.

We use CNN encoders with a fixed-dimension projection head for all pretext tasks except AEs. For AEs, we mirror the encoder to construct a decoder, while the original projection head is a bottleneck. We train AEs with the encoder-decoder structure. We discard the decoder during knowledge distillation.

For CvD and CIFAR-10, we use a ResNet-9 [111] for all tasks. We remove the original classification head from the auxiliary task and use a 128-dimensional head after the pooling layer. We train the students to match the embeddings from this 128-dimensional layer.

We use the embeddings after the final pooling layer of a DenseNet-161 network [112] for the Plant Pathology and X-Ray datasets, resulting in 2208-dimensional embeddings.

3.3 Results

We present the results of the anomaly detectors in §3.3.1 and analyse the properties of the different representations in §3.3.2.

3.3.1 Anomaly detection performance

Tables 3.2 to 3.5 summarise the anomaly detection results per dataset across the configurations. Appendix C.1 contains more detailed breakdowns of the results.

Table 3.2: Cats vs. Dogs results averaged over the two classes. The best anomaly detection results for each representation space are **bolded** and control results are *italicised*.

		Anomaly Detection Method (AUROC)				
Teacher Representation	**Classification Accuracy**	Knowledge Distillation	MSE	Mahalanobis (Diagonal)	Mahalanobis (Full)	Mean
Baseline	*83.40*	***89.37***	*89.04*	*76.12*	*85.93*	85.12
Random	*64.11*	*50.87*	*50.24*	*50.33*	***51.17***	50.65
STL Classification	71.28	61.82	**63.20**	58.80	56.34	60.04
CIFAR Classification	87.57	81.98	**92.29**	78.23	74.91	81.85
STL RotNet	75.87	**54.91**	53.71	52.28	53.96	53.72
CIFAR RotNet	76.65	**56.69**	51.35	51.44	55.85	53.83
CvD RotNet	70.12	**50.18**	49.07	49.29	49.64	49.55
STL AE	64.66	52.03	**52.16**	51.60	51.77	51.89
CIFAR AE	64.11	51.63	**52.06**	51.45	51.10	51.56
CvD AE	64.47	**51.29**	49.93	49.93	50.63	50.45
STL DAE	64.69	**53.65**	51.34	50.89	51.52	51.85
CIFAR DAE	66.13	**53.42**	51.64	51.13	51.50	51.92
CvD DAE	57.23	**50.87**	50.37	50.17	50.66	50.51
Mean	70.02	**58.36**	58.18	55.51	56.53	57.15

Table 3.3: CIFAR-10 results averaged over unimodal and multimodal configurations.

		Anomaly Detection Method (AUROC)				
Teacher Representation	**Classification Acc.**	Knowledge Distillation	MSE	Mahalanobis (Diagonal)	Mahalanobis (Full)	Mean
Baseline	*94.08*	***91.52***	*76.01*	*76.19*	*79.77*	80.87
Random	*40.31*	***55.53***	*54.94*	*54.90*	*54.19*	54.89
STL Classification	56.70	**80.78**	71.24	72.43	73.00	74.36
FMNIST Classification	26.16	**54.34**	53.61	53.68	53.70	53.83
STL RotNet	59.94	**63.01**	61.62	61.41	59.61	61.41
FMNIST RotNet	21.59	**54.68**	52.64	52.93	51.67	52.98
CIFAR RotNet	37.27	**73.35**	72.95	73.07	69.23	72.15
CIFAR SimCLR	65.04	**51.78**	32.52	32.83	47.15	41.07
STL AE	47.13	**57.16**	56.14	56.47	56.29	56.52
FMNIST AE	43.60	**57.69**	55.96	55.64	55.26	56.14
CIFAR AE	44.50	**56.33**	55.91	56.65	55.28	56.04
STL DAE	49.82	55.73	56.92	57.31	**57.68**	56.91
FMNIST DAE	40.57	**56.25**	55.04	55.28	54.26	55.21
CIFAR DAE	14.67	**55.45**	52.95	53.73	54.75	54.22
Mean	45.81	**61.69**	57.75	58.03	58.70	59.04

Table 3.4: Plant Pathology results.

		Anomaly Detection Method (AUROC)				
Teacher Representation	**Classification Acc.**	Knowledge Distillation	MSE	Mahalanobis (Diagonal)	Mahalanobis (Full)	Mean
Baseline	*100*	***100***	*99.87*	*99.89*	*99.89*	99.91
Random	*63.67*	*42.50*	***44.16***	*44.07*	*41.35*	43.02
Plant Village Classification	90.92	89.82	76.57	83.73	**90.78**	85.23
ImageNet Classification	88.24	**70.66**	56.69	61.85	49.45	59.66
Plant Path. RotNet	57.97	**48.09**	46.70	47.14	47.51	47.36
Mean	80.16	**70.21**	64.80	67.34	65.80	67.04

Table 3.5: X-ray results.

		Anomaly Detection Method (AUROC)				
Teacher Representation	**Classification Acc.**	Knowledge Distillation	MSE	Mahalanobis (Diagonal)	Mahalanobis (Full)	Mean
Baseline	*98.98*	***99.74***	*99.25*	*99.68*	*99.63*	99.58
Random	*88.56*	***73.61***	*42.41*	*44.85*	*71.87*	58.19
ImageNet Classification	96.79	**76.36**	68.37	69.35	73.46	71.88
X-ray RotNet	93.62	**96.41**	36.08	37.77	60.06	57.58
Mean	94.49	**86.53**	61.53	62.91	76.26	71.81

3.3.1.1 Knowledge distillation outperforms shallow anomaly detectors

Figure 3.4 summarises anomaly detection performance across all datasets and teacher representations. Knowledge distillation outperforms the other detectors on average, as it makes fewer assumptions about the underlying benign distribution.

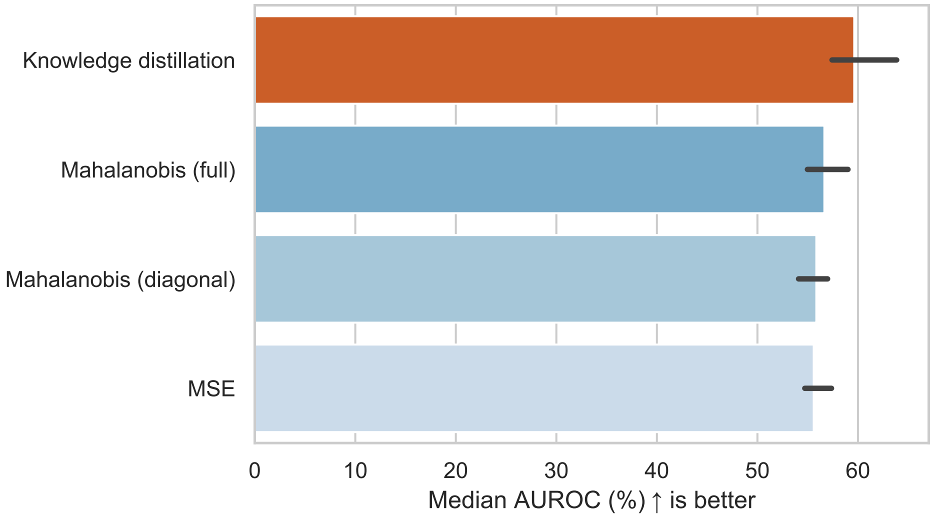


Figure 3.4: Bar chart comparing the different detectors, summarised by all datasets and auxiliary representations.

To understand this in more detail, we can see how knowledge distillation compares to the Mahalanobis distance across different representations on CIFAR-10 (Figure 3.5). We use the Mahalanobis scores for this analysis as performance across the shallow detectors is similar. If knowledge distillation and the Mahalanobis distance were on par, the points should lie on the grey $x = y$ line. This behaviour is the case for the unimodal setting. The detectors are more comparable. However, knowledge distillation has an advantage in the multimodal setting. The results indicate this as the representations skew towards the y-axis.

Mahalanobis distances make assumptions about the teacher embeddings that may not hold. The Mahalanobis distance assumes a multivariate Gaussian distribution models the benign data. MSE also assumes a Gaussian distribution. This assumption may be reasonable for an unimodal setting where the typical class is tighter. However, this assumption may not hold true for the multimodal setting, where we can expect more intra-class variation.

In contrast, knowledge distillation uses the errors between the student and teacher rep-

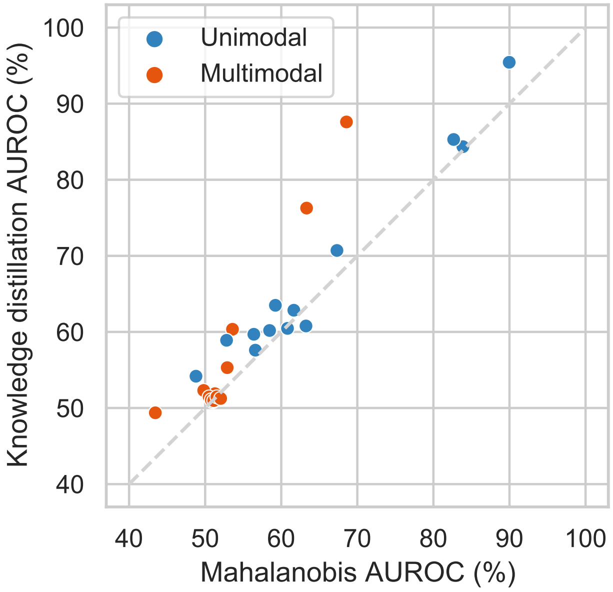


Figure 3.5: Scatter plot of CIFAR-10 Mahalanobis AUROC scores against knowledge distillation AUROC scores, separated by unimodal and multimodal configurations.

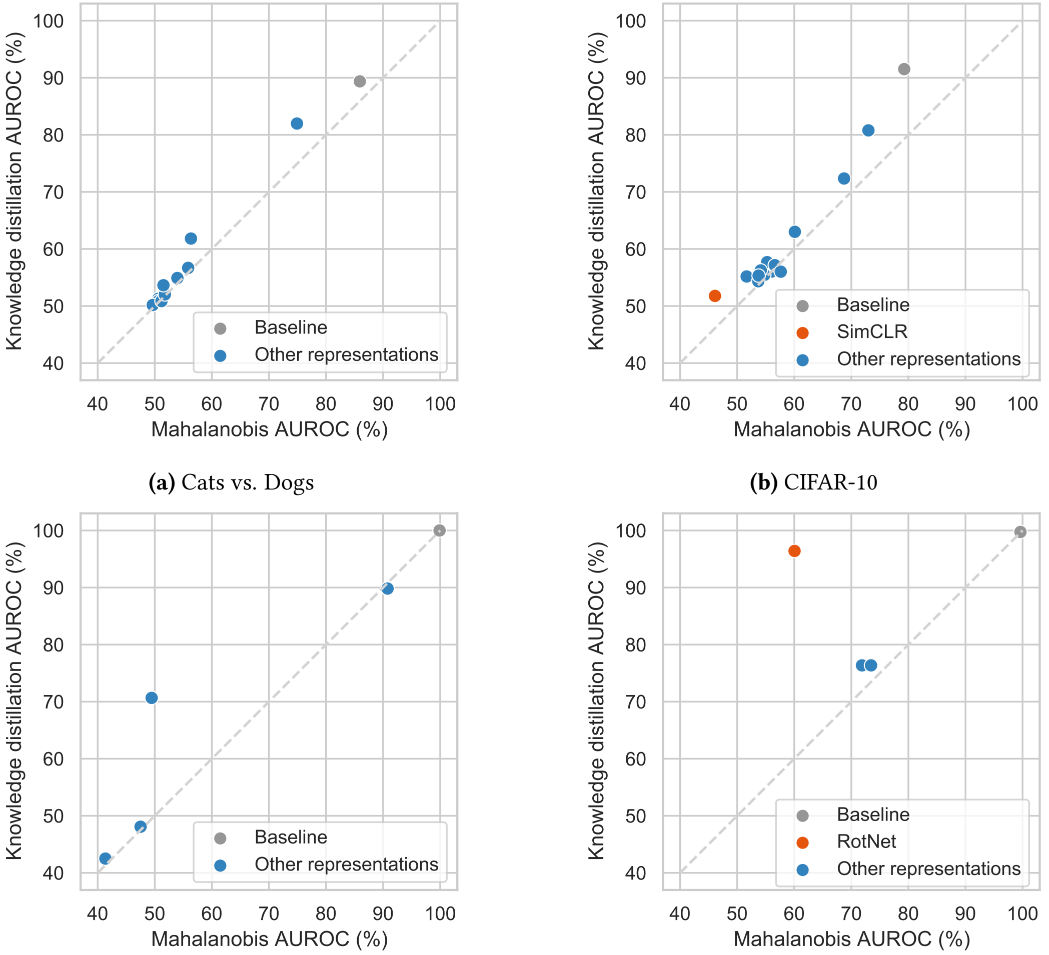
resentations instead of directly using the teacher representations. Hence, there are no explicit constraints on the distribution of the benign class.

Performance amongst the shallow detectors is similar when the datasets involve anomalies from distinct semantic classes (Tables 3.2 and 3.3). Plant Pathology (Table 3.4) and X-ray (Table 3.5), the two datasets where anomalies are closer to the benign distribution, are exceptions. The complete Mahalanobis computation works better than the other shallow approaches. Their results suggest more challenging anomaly detection benefits from considering skewed directions.

3.3.1.2 The choice of representation is more important than the anomaly detector

We expand on Figure 3.5 and analyse embedding performance across the different datasets. Figure 3.6 compares knowledge distillation scores against Mahalanobis scores for each embedding. There is a clear correlation between these scores across the datasets, indicating the choice of representation (and hence embedding) is more important than the anomaly detector.

Discriminative tasks lead to better representations for all of the datasets. For all datasets except for X-ray, transferring features from a classification task containing similar semantic classes is the most beneficial. In the absence of labels, RotNet can help implicitly learn



(a) Cats vs. Dogs

(b) CIFAR-10

(c) Plant Pathology

(d) X-ray

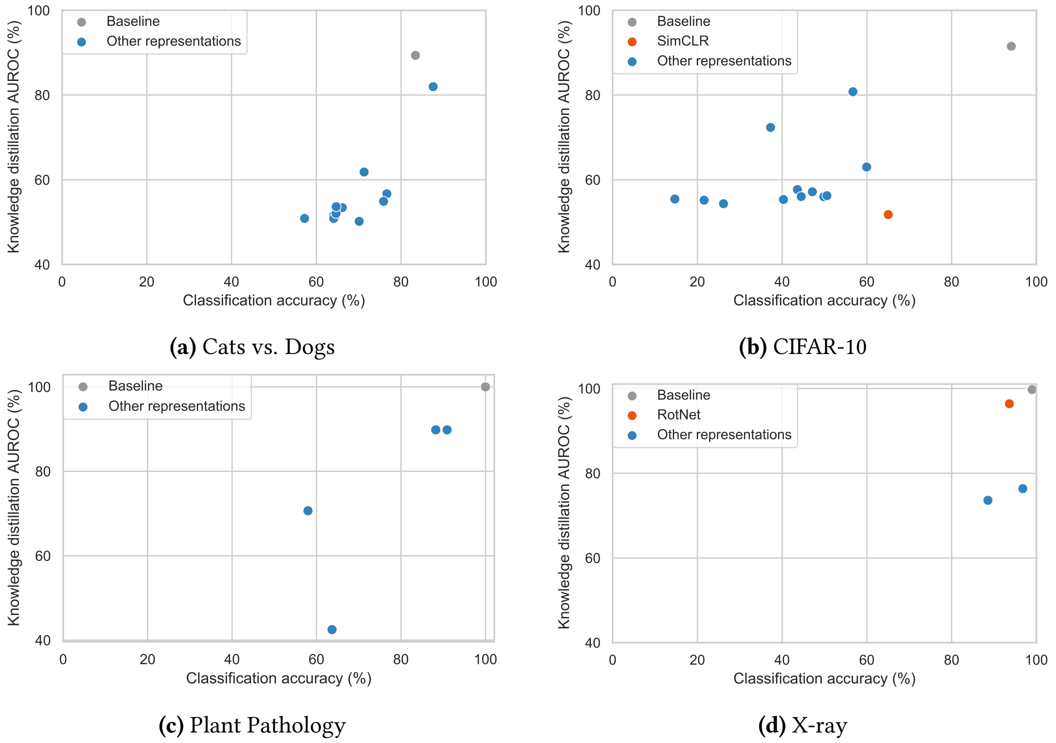
Figure 3.6: Scatter plots of Mahalanobis AUROC scores against knowledge distillation AUROC scores for each representation.

semantic features. These results align with Hendrycks et al. [113], who use RotNet's predictions to detect OOD samples with ImageNet and CIFAR-10 as the benign distributions. However, RotNet is not a general-purpose pretext task. The orientation of leaves is irrelevant to plant health, so RotNet is ineffective for the Plant Pathology dataset.

However, RotNet led to the best results on the X-ray dataset. We achieved an AUROC of 96.41%, exceeding the anomaly detection score of 92.5% in Griffin et al. [17]. There are two potential reasons. Firstly, X-ray images are a different domain than the natural images found in ImageNet. Training on X-ray images directly leads the network to learn features more specific to this domain. Secondly, adversaries might need to fit firearms in specific parcel locations to conceal them. The RotNet task might help identify these positions.

3.3.2 Analysing representational properties

3.3.2.1 Separability is not the sole factor for adequate anomaly detection



(a) Cats vs. Dogs

(b) CIFAR-10

(c) Plant Pathology

(d) X-ray

Figure 3.7: Scatter plots of supervised classification accuracies against knowledge distillation AUROC scores for each representation.

We compare knowledge distillation performance against supervised classification accuracy in Figure 3.7. Although there is a correlation between these values, there are exceptions, which we highlight in the plots. On CIFAR-10, although SimCLR achieves the highest supervised classification performance outside of the baseline (65.04% accuracy), its anomaly detection performance is underwhelming. SimCLR achieves AUROC scores equal to or worse than random. Additionally, although RotNet does not have the highest classification performance amongst the X-ray representations, it is the best for detecting threats.

3.3.2.2 Better representations correlate with higher average gradient norms

Although the anomaly detection results highlight a link with separability, it does not appear to be the sole factor for good anomaly detection. We hypothesise that anomaly detectors also rely on non-robust features. Namely, the distribution of the training data in a suitable representation space has directions in which the distribution is tight. Hence, the features are "brittle". Consequently, these brittle features allow anomalous data to manifest more clearly. Research into adversarial vulnerability indicates these features are essential for good generalisation [99].

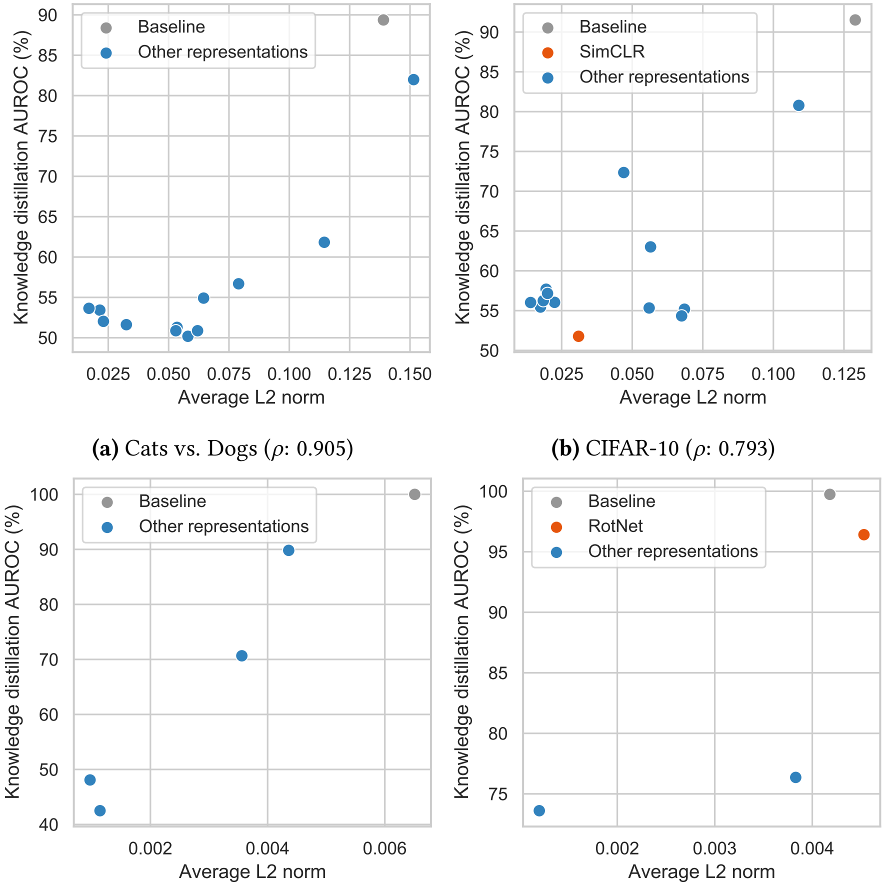
We measure brittleness by adapting the gradient norm approach proposed by Simon-Gabriel et al. [64]. After training a student with the knowledge distillation framework, we record the mean L2 gradient norms of the training data using the student network. To allow for meaningful comparisons of this score across different representations, we divide the norms by the trace of the covariance to account for the spread of differing representations:

$$\frac{\mathbb{E}||\partial_{x_{\text{train}}} L||_2}{\text{tr}(\Sigma_{\text{train}})} \tag{3.2}$$

x_{train} denotes a training sample, L is the MSE loss between the student and teacher and $\text{tr}(\Sigma_{\text{train}})$ is the trace of the training covariance matrix. We compute the covariance using the difference between the student and teacher outputs for each training datum.

More adversarially vulnerable instances require subtler shifts in the input domain to evoke changes in the model's outputs. Hence, a higher gradient norm indicates increased susceptibility to adversarial perturbations [64].

We illustrate the relationship between knowledge distillation performance and the average L2 gradient norms in Figure 3.8. The scatter plots suggest a positive correlation between better performance and higher gradient norms. These norms could also explain SimCLR's performance on CIFAR-10 (as its average L2 norm is lower than other representations) and RotNet's performance on X-ray (as its average L2 norm is high and close to the baseline). These results suggest that higher norms could indicate which representations are better candidates for anomaly detection.



(a) Cats vs. Dogs (ρ: 0.905)

(b) CIFAR-10 (ρ: 0.793)

(c) Plant Pathology (ρ: 0.980)

(d) X-ray (ρ: 0.738)

Figure 3.8: Scatter plots of average L2 gradient norms against knowledge distillation AUROC scores for each representation.

3.4 Conclusion

We conclude by outlining the work's limitations and main contributions.

3.4.1 Limitations

In our experiments, we match the student's architecture with the teacher. This approach does not take advantage of the initial proposed benefits of knowledge distillation. The original paper shows that knowledge from a more complex teacher is transferrable to a more efficient student [87]. In addition, our decision to match the student and teacher architecture limited our experiments, due to constraints on computational resources. For this reason, we were unable to train AEs on Plant Pathology and the X-ray dataset. Future work could establish whether using smaller architectures can maintain anomaly detection performance.

In addition, we did not apply any data augmentations to the input data. As data augmentations can improve separability [101] (which links to anomaly detection performance), detailed ablation studies varying these augmentations could improve anomaly detection performance further.

Our results also show that knowledge distillation outperforms shallow anomaly detectors. However, we only extracted features from the penultimate layer of the teacher networks. Although there is a debate about whether deep anomaly detectors outperform more traditional counterparts [94], extracting embeddings from a later layer may not take advantage of the complete representational power of neural networks. Future work could compare how knowledge distillation compares with embeddings from several layers of the teacher network.

Finally, the gradient norm score is only usable on neural networks that can backpropagate. This score is not suitable for embeddings that do not rely on neural networks or when we do not have white-box access. Extensions to this work should investigate alternative ways to measure embeddings.

3.4.2 Summary

We conducted a study to analyse how various representations affect anomaly detection performance. To do so, we propose a knowledge-distillation framework that fixes all components apart from the teacher representation. We show that knowledge distillation outperforms other parametric methods, especially when the benign distribution is multimodal.

Nonetheless, we show that anomaly detection performance relies on the underlying representation. Our results reinforce previous findings that features from discriminative tasks outperform those from generative models [73]. Transferring features from a similar domain trained on a classification task is most beneficial. Alternatively, discriminative tasks that learn relevant semantic features like RotNet can be substitutes.

Finally, we analyse the properties of the representations. We show separability between anomalies and benign samples is insufficient for reasonable anomaly detection. We propose a gradient norm score that measures the adversarial brittleness of representations, and we link the score to anomaly detection performance.

4 | Text

Monitoring text helps to protect users from harmful or inappropriate online content. There are many uses of text-based anomaly detection, from identifying fake news [114] to weeding out spam [115] to flagging atypical reviews [116],

Existing works in natural language processing (NLP) focus on the far OOD setting, in which the anomalies derive from a dataset that has a different purpose [13, 117, 118, 119, 120]. For example, a model could use news articles as training data but treat film reviews as anomalous [120]. These approaches also assume the training data contains multiple subclasses. As a result, the anomaly scoring mechanisms typically incorporate these supervisory signals by fitting a Mahalanobis distance [96] to each subclass or by obtaining the highest probability in the softmax layer [13]. However, these supervisory signals are not always available. Social media posts, for instance, may cover a range of topics. Hence, it might be challenging to assign definitive labels.

One-class learning removes the need to annotate subclasses. However, compared to images, fewer studies investigate the effectiveness of one-class learning on text [66, 80, 113]. Recent innovations in machine learning suggest that adapting image-based approaches to text might be of benefit. Previously, image and text tasks used different architectures. CNNs were the architecture of choice for images, while recurrent neural networks (RNNs) were preferable for text [101]. The emergence of transformers has altered this setup. Transformers have become the architecture of choice for image and text-based tasks [82, 121], outperforming the previous go-to architectures.

Reconstruction-based methods are a common anomaly detection approach for images [71, 72]. They assume a model trained on benign data cannot represent unseen anomalies well. Hence, the reconstruction losses for anomalies should be higher. We investigate whether the same principle is suitable for textual anomaly detection. We fine-tune transformers on benign text data using three self-supervised objectives and use the loss as the anomaly score. Focusing on the near OOD anomaly setting, we examine performance on two anomaly types with six datasets. We refer to the two anomaly types as semantic and word order. Semantic anomalies are created by partitioning a single dataset by class label, while word-order anomalies randomly shuffle the tokens of benign text. Our approach outperforms more complex methods, boosting the average AUROC score on semantic anomalies by 11.6% and word-order anomalies by 22.8%. Our contributions are as follows:

- We show that fine-tuning a transformer is a simple benchmark that achieves good anomaly detection results and outperforms other approaches.
- We introduce word-order anomalies to measure how sensitive an anomaly detector is to word-order information.
- We show the optimal self-supervision objective and the resulting representation depends on the anomaly type. Separability of anomalies and benign data is a necessary but insufficient condition. Adversarially brittle features are better for detecting word-order anomalies.

We provide a background into textual OOD detection in §4.1. We outline our method in §4.2 and analyse the results in §4.3. §4.4 concludes the chapter and identifies areas of future work.

4.1 Background

This section discusses existing anomaly and OOD detection approaches for text.

4.1.1 Anomaly detection with static word embeddings

Previous textual anomaly detection approaches relied on feeding static word embeddings into shallow detectors [122]. Static word embeddings are functions that map each word to a single vector [123]. These word embeddings can be simple word frequencies [123] or more sophisticated learnt dictionaries like word2vec [124], GloVe [125] and FastText [126]. However, static approaches fail to capture nuance. They map words with multiple meanings to the same embedding [123].

As a result, anomaly detection approaches shifted to learning word representations specifically for OOD detection. **Context vector data description (CVDD)** is one example [116]. Starting from a pre-trained word embedding method like FastText, CVDD uses multiple self-attention heads [127] to learn a collection of context vectors. The method applies an orthogonal regularisation term to ensure the context vectors learn different concepts. Context vectors are "prototypical centres": compact descriptions of the various concepts in benign data. A context vector is like the centre embedding in SVDD [65, 66]. During training, the model also learns to map input sentences to the new representation space. The idea is that benign data maps to a location close to the context vectors, whereas anomalies lie far away. Therefore, the cosine distance between an input sentence and a context vector can evaluate anomalousness. Hence, the overall anomaly score is an average cosine distance between the input and each context vector.

Because CVDD maps data to a central embedding, the authors mention manifold collapse could occur. The authors state this only happens if a word appears in an identical location for all training samples. They mitigate the collapse issue by normalising the context vectors. CVDD contains several hyperparameters. In addition to the number of attention heads, the output dimensionality, context regularisation term, and importance weighting between different heads can also influence results. Balancing these hyperparameters can make it challenging to deploy CVDD.

4.1.2 Multiple class out-of-distribution detection methods

There are more existing works that study text-based OOD detection. However, they rely on subclass labels. We outline a few examples below.

Hendrycks et al. [128] show that fine-tuning a pre-trained transformer on a classification task and using the maximum softmax probabilities [13] to score anomalies outperforms shallow detectors trained with static embeddings. However, they use far OOD datasets for evaluation, which are more trivial to address.

Arora et al. [117] extend the experiments in Hendrycks et al. [128] to more near OOD settings. They analyse two types of near OOD scenarios: background shift and semantic shift. Background shifts include more syntactical changes, such as spelling errors. Semantic shifts involve topic changes. They compare two OOD scorers: maximum softmax probabilities [13] against perplexities. However, they use different backbones for the scoring methods: RoBERTa [83] for maximum softmax and GPT-2 [129] for perplexities. Therefore, performance differences could be due to a combination of the scoring method and the underlying architecture.

k-Folden [118] extends the maximum softmax method in Hendrycks et al. [128] by simulating proxy anomalies. Given a dataset with k subclasses, they train k models. For each model, they leave one class out and train the model to classify the remaining $k - 1$ classes with a cross-entropy loss. The left-out class is the proxy anomaly class. They supplement the cross-entropy loss with a Kullback-Leibler (KL) divergence loss between the proxy anomalies and a uniform distribution. To assess anomalousness, they look at the softmax probabilities of the left-out class for each model and average the probability distributions. Similar to vanilla maximum softmax, on average, this mean score should be lower for OOD data because the models are more uncertain. However, ensembling is more expensive than alternative methods.

Podolskiy et al. [119] revisit the Mahalanobis distance [96]. Like Hendrycks et al. [128], they fine-tune transformers on a classification task. However, they use class-wise Mahalanobis distances as anomaly scores and find this outperforms maximum softmax probabilities.

4.1.3 One-class anomaly detection methods

The methods in the previous section rely on subclasses. They also only use the classification layer to identify anomalies. Xu et al. [81] argue that this does not incorporate the full representational power of transformers. They propose **Mahalanobis distance as features (MDF)** as a solution. MDF computes the Mahalanobis distance of each training datum in each layer. They use these distances as features to train an OCSVM. They find that fine-tuning the models with a masked language modelling loss before feature extraction also improves results. However, their investigations on the best intermediate layer for anomaly detection are inconclusive. No particular layer is better than the others. Although training OCSVMs with Mahalanobis distance features is computationally efficient, the distances are less informative than the complete representation space.

The closest approach to ours is **Detecting Anomalies in Text using ELECTRA (DATE)**

[130]. DATE is an end-to-end approach that adapts the ELECTRA architecture [131]. ELECTRA is a large language model (LLM) with the same transformer-encoder architecture as BERT [82]. However, it uses a replaced token detection (RTD) objective for learning instead of masked language modelling. In RTD, an additional BERT-based generator replaces a proportion of input tokens with plausible random tokens. The main component of ELECTRA, the discriminator, predicts whether the tokens in an input sequence are original or inserted by the generator. DATE incorporates RTD and an additional objective called replaced mask detection (RMD). For the RMD task, the discriminator chooses which mask pattern corrupted the text from a set of potential masks. Inference only uses the discriminator. The discriminator receives an uncorrupted sentence. It then calculates the probability of each token being original using the RTD head. Like the maximum softmax score [13], the RTD head should be less confident about anomalies.

4.2 Method

Our review of text-based OOD detection suggests few works combine representation learning with one-class detectors. The variations in self-supervised objectives across approaches also make it unclear what objective and resulting representation is best for anomaly detection [117, 130]. Therefore, our research questions are as follows:

1. What self-supervised objective works best for anomaly detection on text?
2. What properties do the resulting representations have?

4.2.1 Overall principle

We fix the underlying architecture to minimise the influence of architectural differences and only vary the self-supervised objectives. In particular, we use the encoder from a pre-trained uncased BERT$_{\text{BASE}}$ [82] and RobERTa$_{\text{BASE}}$ [83]. We use the uncased versions to ensure a fair comparison with other approaches like CVDD [116], which lowercase all data as a pre-processing task. We also initialise from pre-trained weights to benefit from the transfer learning strengths of LLMs. We append different heads depending on the objective. We fine-tune each model for up to 30,000 steps and employ early stopping based on the validation loss. We analyse three self-supervised objectives in our experiments.

We choose an end-to-end anomaly detection approach to use the entire representational capacity of LLMs. We assume that a fine-tuned model learns the underlying characteristics of benign data well but not those of anomalies. This assumption is analogous to using autoencoders for image-based anomaly detection [45, 46, 71, 72]. The reconstruction error for anomalies should be higher because an autoencoder only contains the features to construct benign data. Consequently, our approach uses the loss as the anomaly score.

We choose to use the loss of the encoder directly rather than the student-teacher disagreement described in Chapter 3. We disregard this approach as the student-teacher method requires an additional frozen model.

4.2.2 Self-supervised objectives

We analyse three self-supervised objectives in our experiments:

Masked language modelling (MLM) is a perturbation-type objective [82]. It is essentially a denoising autoencoder [102]. MLM involves randomly masking tokens in a sequence and training the model to predict the masked tokens. Through this approach, the model learns the contextual relationships between words. Both BERT and RoBERTa use a fixed `[MASK]` token to mask inputs [82, 83]. When fine-tuning our models, we retain the default configuration specified in the original BERT implementation and randomly mask 15% of tokens.

The loss function only considers the masked tokens. Therefore, if the input sequence has n tokens, where the i-th token is masked (denoted by y_i) and the other tokens $(x_1, x_2, ..., x_{i-1}, ..., x_{i+1}, x_n)$ are provided in the sequence, the MLM loss at position i is as follows:

$$\mathcal{L}_{MLM} = -\sum_{i=1}^{n} \log(p(y_i | x_1, x_2, ..., x_{i-1}, \texttt{[MASK]}, x_{i+1}, ..., x_n)). \tag{4.1}$$

At inference, we mask the same proportion of tokens in the test sentence and use the error between the predicted and original tokens as the anomaly score.

Causal language modelling (CLM) is a conditional prediction task. The model learns to predict the next token, given previous tokens in a sequence. Unlike MLM, it is unidirectional as it only considers tokens before the token of interest. However, CLM-based models exhibit good downstream performance on language tasks [129]. They have shown strong performance on few-shot learning tasks like translation, question-answering and on-the-fly reasoning [132]. The loss at token position i is as follows:

$$\mathcal{L}_{CLM} = -\sum_{i=1}^{n} \log(p(\boldsymbol{y_i}|\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x_{i-1}})) \tag{4.2}$$

We use the CLM loss as the anomaly score. The CLM loss closely links to perplexity, which is the exponential of the CLM loss. Perplexity is a frequent metric for evaluating language models [123]. Lower perplexities mean the model's predictions are closer to the actual distribution of the data. Other works have used perplexity to evaluate far OOD detection [117] and evidence-supported fact-checking [114]. However, we assess near OOD anomalies and do not combine the test sequence with supporting evidence to identify anomalies.

Contrastive learning refers to a collection of clustering methods with learnt distance mappings. It encourages similar instances to group closely and to lie far from disparate samples. Previous works in computer vision suggest contrastive losses can help discriminate anomalies from benign samples [133, 134]. However, these methods require data augmentations that are not directly transferrable to text, such as colour jittering. Zhou et al. [120] propose a contrastive pre-training method for text-based OOD detection but use subclass labels to define positives and negatives.

SimCSE [135] resolves the data augmentation issue for unsupervised text data by applying different dropout masks to sentences. The model learns to select the same (perturbed) sentence from a minibatch of other sentence pairs. We fine-tune the model using the original dropout probability ($p = 0.1$) and temperature ($\tau = 0.05$) described in SimCSE. We use the loss, also known as NT-Xent [40], as the anomaly score.

For a feature encoder f_θ and a random dropout mask $\boldsymbol{z}$, the encoding for a sentence $\boldsymbol{x_i}$ is $\boldsymbol{h}_i^{\boldsymbol{z}}$. Therefore, if we feed $\boldsymbol{x_i}$ to the encoder twice and apply two different dropout masks $\boldsymbol{z}$ and $\boldsymbol{z}'$, the outputs are $\boldsymbol{h}_i^{\boldsymbol{z}}$ and $\boldsymbol{h}_i^{\boldsymbol{z}'}$. For a minibatch of N sentences, and some similarity function sim(.), the loss at $\boldsymbol{x_i}$ is as follows:

$$\mathcal{L}_{NT-Xent} = -\log\left(\frac{\exp^{\mathrm{sim}(\boldsymbol{h}_i^{\boldsymbol{z}}, \boldsymbol{h}_i^{\boldsymbol{z}'})/\tau}}{\sum_{j=1}^{N} \exp^{\mathrm{sim}(\boldsymbol{h}_i^{\boldsymbol{z}}, \boldsymbol{h}_j^{\boldsymbol{z}_j'})/\tau}}\right) \tag{4.3}$$

4.2.3 Comparisons

We analyse the three fine-tuned objectives against five baselines. We chose the baselines as they were introduced in previous text anomaly detection work. In addition to **CVDD** [116], **MDF** [81] and **DATE** [130], we use the following for comparison:

Pre-trained transformers (Pre-trained). We use the MLM loss as the anomaly score on a pre-trained network. This is equivalent to using the pre-training distributions, BooksCorpus [136] and Wikipedia, as the benign distribution. Although this approach was not included in previous work, it is comparable to MLM. We can compare it to MLM to examine the incremental benefit of fine-tuning. We disregard the auxiliary next-sentence prediction in BERT [82] as we do not use sentence pairs for anomaly detection.

Bag-of-words models (BoW). We follow the approach in CVDD [116] and compute the mean over word embeddings extracted from FastText [126] to create a sentence embedding for each datum. We use these sentence embeddings to train linear OCSVMs. We do not use pre-trained embeddings from BERT as Ruff et al. [116] observed they did not garner enough improvements to justify the additional computational cost. By including BoW, we can assess whether dynamic word embeddings that consider sentence structure are better than using word frequencies.

4.2.4 Datasets

We evaluate anomaly detection performance on the same publicly available datasets used in previous NLP anomaly detection work.

20 Newsgroups [137] is a collection of 20,000 documents split across 20 different newsgroups. We use the six top-level subjects (*computer, recreation, science, miscellaneous, politics, religion*) to partition the documents. For each subject, there are 577-2,859 training samples and 382-1,909 test samples.

Reuters-21578 [138] is a collection of 10,788 news articles split across 90 topics. We only use a subset of data that have only one label (*earn, acq, crude, trade, money-fx, interest, ship*). Partitioning by class label, there are 108-2,840 training samples and 36-1,083 testing samples.

AG News [139] is a topic classification dataset gathered from more than 2,000 news sources over one year of activity. It contains four classes (*business, sci, sports, world*), each with 30,000 samples for training and 1,900 for testing.

IMDb [140] is a sentiment classification dataset consisting of film reviews. It contains two classes (*pos, neg*), each with 25,000 samples for training and 25,000 for testing.

Snopes [141] is a fact-checking dataset containing paired examples of tweets and a fact-checking article from *snopes.com*. There are four classes (*true, mostly true, mostly false, false*). We only use *true* (7,363) and *false* (21,256) tweets in our experiments and do not use the articles. We randomly partition 80% of this smaller dataset for training and use the remaining 20% for testing.

The **Enron Spam Dataset** [142] is derived from the Enron Email Dataset [143]. There are two classes, *ham* (16,458) and *spam* (17,171) emails. We randomly partition 80% of the dataset for training and use the remaining 20% for testing.

We pre-process all data in the same manner as CVDD [116] for a fair comparison. Namely, we lowercase all sentences and strip punctuation, stopwords, numbers and whitespaces. We also only include words with a minimum length of three characters.

4.2.5 Anomaly construction approach

We use the class labels of the datasets to construct two setups for the benign training data, as per Kim et al. [46]. This setup allows us to compare anomaly performance between inliers having a tighter or more diverse distribution. Therefore, for a dataset with m class labels, we make the following arrangements:

- **Unimodal normality**: We construct the inliers using data from a single label.
- **Multimodal normality**: We construct the inliers using data from $m-1$ labels.

We study two types of near OOD anomalies:

Semantic anomalies. Data belonging to the same original class label(s) as the training data are categorised as benign, while the remainder are anomalies. Taking AG News as an example, if we selected *business* as the unimodal benign class, test sentences belonging to *business* would be benign. Anything from *sci*, *sports*, and *world* would be anomalies. Conversely, if we used *sci*, *sports*, and *world* as multimodal benign samples, test sentences from *business* would be anomalies.

Word order anomalies. The anomalies have the same semantic content as the benign samples but have scrambled word order. We partition the text into non-overlapping n-grams ($n \in \{1, 2, 3, 4\}$) and shuffle until each n-gram is no longer in its original position. Larger values of n are more similar to the benign class and more challenging to detect. If we use the AG News *business* category as the unimodal benign class, the anomalies would be perturbed sentences from the test split of *business*.

We use the same seeded random function algorithm described in Sinha et al. [144], who find that masked language models pre-trained with perturbed word order still achieve high accuracy when fine-tuned for downstream tasks. We use word order anomalies to disentangle whether detectors focus on word frequency statistics or if they also consider sentence structure at inference. Therefore, approaches unable to detect word order anomalies are more similar to bag of words models.

We provide examples of word order anomalies below:

Class	**Sentence**
Inlier	voip gaining ground despite cost concerns
Anomaly ($n = 1$)	concerns voip despite cost ground gaining
Anomaly ($n = 2$)	ground despite cost concerns voip gaining
Anomaly ($n = 3$)	despite cost concerns voip gaining ground
Anomaly ($n = 4$)	cost concerns voip gaining ground despite

Table 4.1: Word order anomaly examples derived from AG News. Colour corresponds to the original order.

4.3 Results

We present the results for semantic and word order anomaly detection. We examine how architectural choices influence results and then analyse the properties of the learnt representations.

4.3.1 Semantic anomaly detection

Figure 4.1 shows the overall anomaly detection results for semantic anomalies. The results labelled pre-trained, MLM, CLM and SimCSE refer to the $\text{BERT}_{\text{BASE}}$ models. The full results split by dataset and normality, including $\text{RoBERTa}_{\text{BASE}}$, are in Appendix C.2.

We only highlight $\text{BERT}_{\text{BASE}}$ results as the $\text{BERT}_{\text{BASE}}$ and $\text{RoBERTa}_{\text{BASE}}$ results are com-

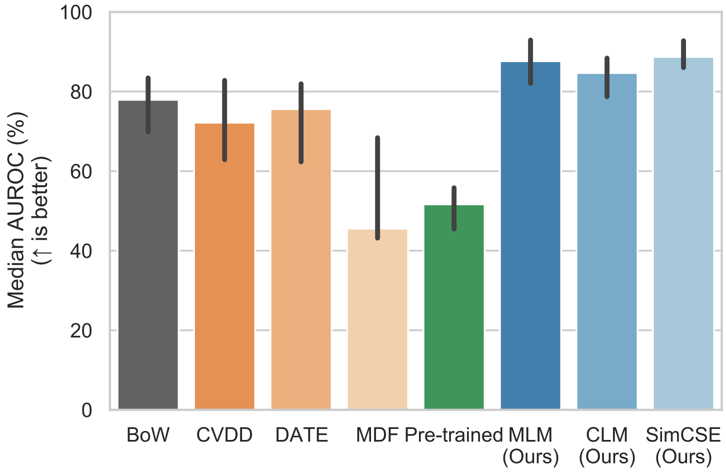


Figure 4.1: Median semantic anomaly detection results aggregated by model. Error bars denote the 95% confidence intervals across all datasets.

parable. Therefore, we display results for BERT$_{\text{BASE}}$ unless otherwise stated. We compare BERT$_{\text{BASE}}$ and RoBERTa$_{\text{BASE}}$ in §4.3.3.1.

4.3.1.1 Fine-tuning a pre-trained transformer boosts semantic anomaly detection performance

Figure 4.1 indicates BoW can detect semantic anomalies adequately. These results suggest semantic anomalies are detectable by analysing word frequency statistics. In contrast, pre-trained BERT is not much better than random. The underlying representation relies on a general-purpose corpus rather than conditioned on the content of the benign class, so it does not capture class-specific word frequency statistics.

The improvements seen with MLM, CLM and SimCSE suggest fine-tuning helps to give additional information about the nature of benign data. This observation aligns with findings in OOD detection for images [145]. For images, the authors find that fine-tuning transformers outperform pre-trained variants and speculate this is due to the model learning more tightly clustered benign embeddings.

When comparing MLM, CLM and SimCSE, their performances are on par. The exception is IMDb (Appendix C.2, Figure C.1i). SimCSE exceeds the other approaches. It could be because the semantic content of positive and negative reviews are similar, but SimCSE is better at capturing the overall nuance of sentences. The loss function in SimCSE, NT-Xent, considers the entire sentence representation.

4.3.1.2 SimCSE is more robust to contaminated training data

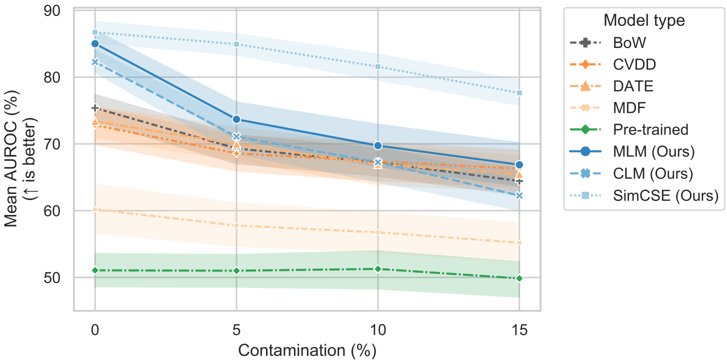


Figure 4.2: Mean AUROC scores across datasets by contamination percentage. Experiments were conducted with semantic anomalies. Pre-trained, MLM, CLM, and SimCSE refer to $\text{BERT}_{\text{BASE}}$ models.

One-class methods assume training data only includes benign samples. However, anomalies may leak into the training data in practice. We simulate this scenario by adding a set percentage of semantic anomalies $\{5\%, 10\%, 15\%\}$ into the training data. On average, the fine-tuned transformers perform better than the other models (Figure 4.2). Contamination affects word frequencies. SimCSE's ability to capture nuance makes it the most robust approach, while CLM is more sensitive.

4.3.2 Word order anomaly detection

We look at overall word order anomaly detection performance and ablations by *n*-gram.

4.3.2.1 Fine-tuning transformers boosts word order anomaly detection

Figure 4.3 shows the overall results on word order anomalies. Under this scenario, BoW performance drops significantly. We also observe these trends in CVDD and DATE, which suggest these methods also mostly rely on word frequency statistics. In contrast, the pre-trained transformer detects word order anomalies better than the other baseline models. As CVDD, DATE and the pre-trained BERT model all include attention mechanisms in their architecture, it suggests sensitivity to word order has more to do with the training objective than the architecture.

Fine-tuned MLM and CLM improve on the performance of pre-trained BERT. However, SimCSE performance is no better than random because it evaluates entire sentences, not individual tokens. Therefore, it cannot deduce shuffled word order. These results indicate

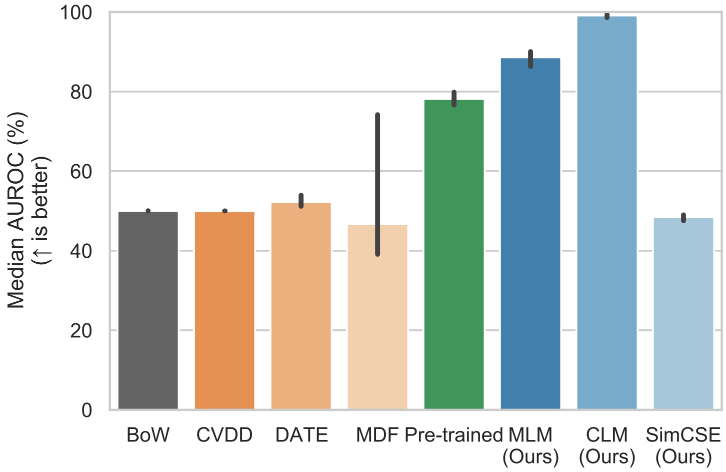


Figure 4.3: Word order anomaly detection results aggregated by model. Error bars denote the 95% confidence intervals across datasets.

the self-supervision approach is more important than the architecture.

4.3.2.2 Density models are better at detecting word order anomalies

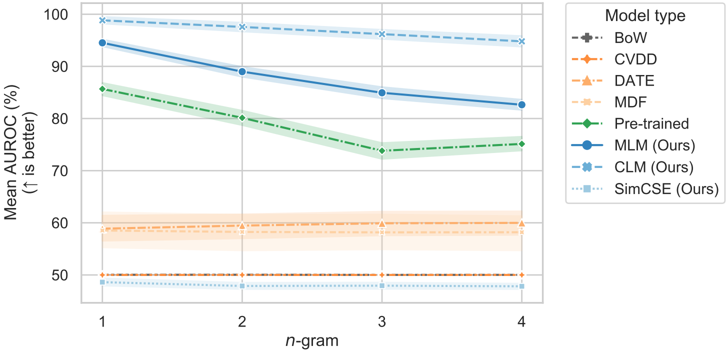


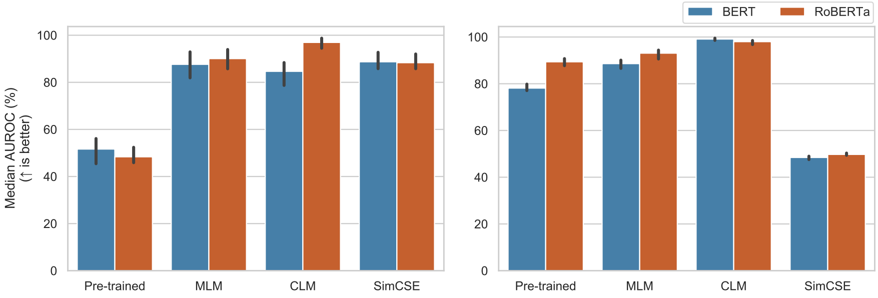
Figure 4.4: Mean AUROC across datasets on word order anomalies by n-gram level. Larger n-grams are more challenging to differentiate from benign samples as fewer individual tokens are shuffled.

We conducted an ablation study of word order detection performance under different permutation strengths. CLM is more stable under more challenging anomaly detection conditions (Figure 4.4), experiencing a decline of only 4% between 1-grams and 4-grams. Pre-trained and fine-tuned MLM experience similar drops (11%), which confirms the choice of objective for anomaly scoring is a core component of performance. As CLM calculates its score at the token level, it is more sensitive to word order changes than MLM, which considers token spans through the masking mechanism.

4.3.3 Architectural choices

We proceed to examine how changes to the architecture influence anomaly detection.

4.3.3.1 Fine-tuning reduces the advantages of better pre-trained architectures



(a) Semantic anomaly results.

(b) Word order anomaly results encompassing all n-grams.

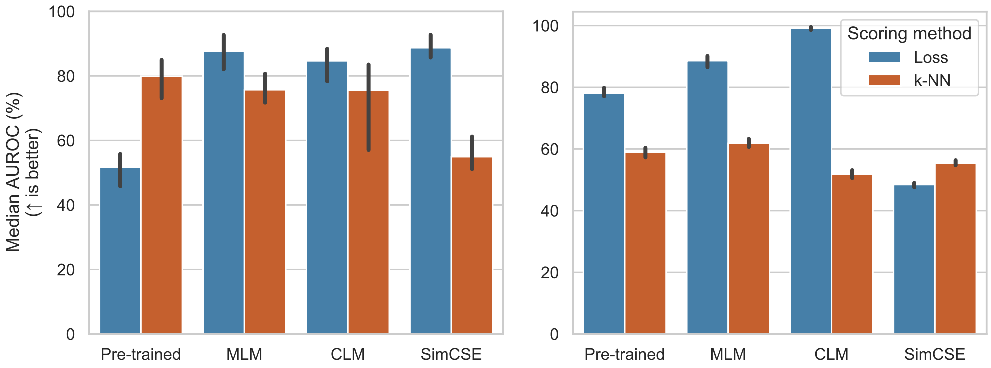
Figure 4.5: Median anomaly detection score for BERT and RoBERTa models across datasets.

Figure 4.5 compares the fine-tuning results on the $\text{BERT}_{\text{BASE}}$ architecture against the $\text{RoBERTa}_{\text{BASE}}$ architecture. Although pre-trained RoBERTa performs better than its BERT counterpart (suggesting RoBERTa is more sensitive to word order structure because it has been pre-trained for longer and with a better masking scheme), this advantage decreases upon fine-tuning. There are also no distinct differences in the semantic anomaly scenario.

4.3.3.2 Using the loss combined with the embedding is better than using the embeddings as a feature extractor

For these analyses, we extracted the embeddings at the last hidden BERT layer (layer 12) and mean-pooled over the positions to analyse the characteristics of the learnt embeddings. We used mean pooling as it is a strong baseline in other NLP tasks like sentence similarity [146]. We process test data in the same way. At inference, we pass a test datum through the fine-tuned transformer and mean-pool over the positions again to extract the embeddings.

We used the embeddings to train static detectors, including OCSVMs, Mahalanobis distance and k-NN. However, we found that k-NNs performed best overall. We report the remaining results in Figure C.3. We use the mean distance from the test datum to its k-nearest neighbours as the anomaly score. We tried $k = \{1, 2, 5, 10, 50\}$ and found $k = 1$ was the best-performing setting.



(a) Semantic anomaly results.

(b) Word order anomaly results encompassing all n-grams.

Figure 4.6: Comparison between using the loss as an anomaly score and nearest neighbour for anomaly detection across datasets.

Figure 4.6 compares the median anomaly detection AUROC scores when using the models end-to-end compared to 1-NN. Although the embeddings themselves are generally capable of supporting semantic anomaly detection, all representations receive a performance boost when using the loss to score anomalies. The embeddings are adequate in the semantic anomaly setting but underperform on word order anomalies. This behaviour suggests that fixed embeddings only consider word frequencies, not sentence structure.

The results also explain why MDF underperforms the other methods (Figure 4.1 and 4.3), as it extracts features from frozen hidden layers to train anomaly detectors.

4.3.4 Analysing representational properties

We proceed to analyse the properties of the self-supervised representations. We train linear classifiers to set a score ceiling and examine properties using gradient norms.

4.3.4.1 The separability of benign data and anomalies matters more for semantic anomalies than word order anomalies

We extracted both benign and anomalous embeddings at the last hidden state of BERT and trained a logistic classifier to examine the separability of the embeddings. As it is supervised, the logistic classifier serves as an upper bound for anomaly detection performance. The correlation between classification accuracy and anomaly detection is more apparent for semantic anomalies (Figure 4.7), suggesting separability is a good indicator for better embeddings for this type. There is no such correlation for word order anomalies.

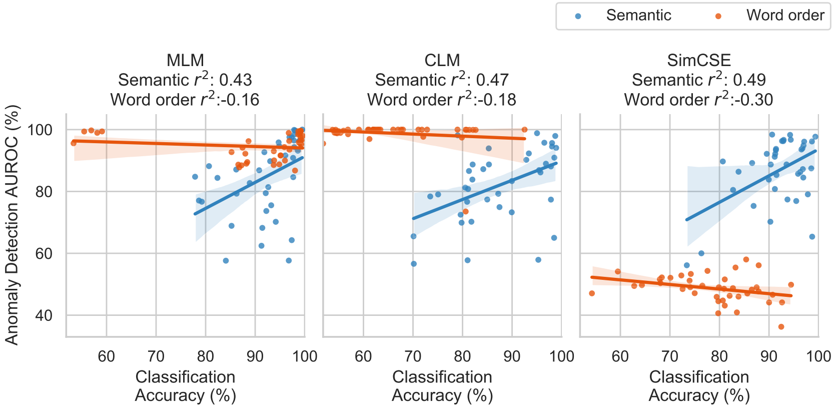


Figure 4.7: Scatter plot comparing classification accuracy of test inliers versus anomalies to anomaly detection performance across datasets.

For instance, some MLM and CLM embeddings have high anomaly detection performance but low classification separability. In addition, SimCSE word order embeddings can be linearly separable, but anomaly detection is ineffective. These patterns suggest there is another factor that influences word order anomaly detection.

4.3.4.2 Word anomaly detection links to the presence of non-robust features

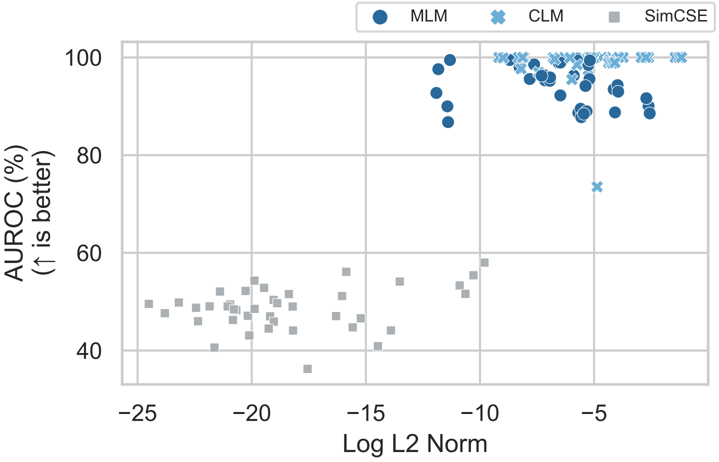


Figure 4.8: Comparison of average log L2 norms of the training inlier data to 1-gram word order anomaly detection performance across datasets. The pattern is similar across different *n*-gram levels.

We hypothesise that an adversarially non-robust [99] inlier embedding is a better signal for word order anomaly detection than separability. Non-robust embeddings are more likely to shift when there is a minor change in the input features. Such embeddings characterise inliers narrowly and provide more directions for anomalies to manifest. We use the same approach as §3.3.2.2 for image-based anomaly detection. We calculate the average L2 gradient norms of the losses divided by the trace of the training data. Higher values correspond to the presence of more non-robust features. Figure 4.8 shows that CLM-based embeddings (which have the best performance) tend to have higher values and SimCSE the least. These clusters correspond with previous literature that states autoregressive models like GPT [129] are highly anisotropic [147], while contrastive models like SimCSE are more isotropic [135, 148].

4.4 Conclusion

We conclude by outlining the work's limitations and main contributions.

4.4.1 Limitations

Our experiments aim to evaluate if transformers fine-tuned with self-supervision objectives can perform anomaly detection. We achieve this by using the same hyperparameters used to pre-train the original BERT and the SimCSE implementations. These hyperparameters may not be the optimal configuration for anomaly detection as the datasets we examined were smaller compared to the original pre-training datasets and differed in content. An extension of this work could look at varying these hyperparameters. Variations could include using different masking rates for MLM [149], investigating alternatives to uniform masking for MLM [150, 151], or adjusting the temperature parameter for SimCSE.

Similarly, we used the same data pre-processing approach described in CVDD to ensure comparability between the methods. Including casing, numbers, or punctuation might affect detectability. For example, the cased version of BERT performs better on some downstream tasks like named entity recognition [152]. Additional ablations could compare uncased BERT with cased BERT or amend the data processing pipeline to analyse performance changes.

In addition, our current approach relies on fine-tuning all layers in a transformer. This step is computationally expensive and may be impractical when computational resources are more limited. Future work could examine alternatives, such as fine-tuning the latter layers only or using adapters [153].

The comparisons to the shallow detectors in §4.3.3.2 are limited. We only extract embeddings from the final hidden layer of BERT, but this approach does not consider the entire representational capacity of the models. Although summarised by the Mahalanobis features, MDF, for example, extracts features from multiple layers [81]. A more detailed comparison could evaluate different ways of combining the features and how this affects results.

Moreover, although word order anomalies demonstrate that fine-tuned transformers are better than BoW detectors, their practical application is unclear. Additional ablations could examine how syntactical changes like spelling errors or negotiations affect detection.

4.4.2 Summary

We studied how fine-tuned transformers with three self-supervised objectives perform anomaly detection. Fine-tuning a pre-trained transformer allows the model to learn better representations than static embeddings. We also show that the output of the loss is a better detection method than extracting the embeddings to train shallow detectors. The best self-supervised objective depends on the type of anomalies. CLM is better at discerning discrepancies in word order, whereas SimCSE is better at capturing nuances in entire sentences. Future work could add outlier exposure to encode more information about anomalies [154]. Alternatively, in situations where the nature of anomalies is unknown, studies could combine self-supervised objectives to complement each other.

5 | Speech: human-level deepfake detection

Audio-based anomaly detection has a wide range of security applications. Potential use cases range from gunshot detection [155] to monitoring machine failures [156, 157, 158] to locating cases of animal distress [159]. However, we focus on harms that directly impact people. The following two chapters focus on speech deepfakes and developing systems to identify them.

Speech deepfakes are artificial voices generated by machine learning models. Due to rapid research progress, it is possible to produce a realistic-sounding clone using only a few audio samples. This development raises the prospect of exploiting speech deepfakes for various criminal activities. Examples include spear phishing, propagating fake news, and bypassing biometric authentication systems [160, 161, 162].

Previous literature has highlighted deepfakes as one of the biggest security threats arising from artificial intelligence progress due to their potential for misuse [161, 162]. For example, experts expect disinformation from deepfakes to erode trust on several levels: towards individuals, organisations, and even societies.

Although several studies examine deepfake performance, they focus on images and videos. Fewer works concentrate on speech data. Therefore, understanding the risks of speech deepfakes will enable the development of better defences and regulations to counteract hazards before they occur.

In this chapter, we measure how well humans can detect speech deepfakes, and was published in PLoS ONE [8]. Estimating human performance allows us to quantify how likely deepfakes can fool individuals. The results give a baseline for how well automated detection solutions need to work to protect against harm. Chapter 6 investigates how different representations affect automated anomaly detection performance.

5.1 Introduction

Adversaries are already using speech deepfakes to commit fraud. In 2020, a bank manager in Hong Kong received a phone call from someone sounding like a company director he had spoken to before [5]. The purported director requested the bank manager to authorise transfers totalling $35 million. Based on their existing relationship, the bank manager transferred $400,000 before he realised something was wrong. The bank manager was a victim of an elaborate hoax: fraudsters had used deepfake technology to clone the director's voice. This incident is not isolated. In 2019, the CEO of a UK-based firm was swindled by a speech deepfake of his manager into transferring €220,000 to a Hungarian supplier [6].

Existing speech deepfake detection research focuses on developing machine learning systems in the context of voice authentication [163, 164, 165]. Comparisons beyond biometrics and studies which measure human detection capabilities are sparse [166].

The state of existing research raises questions. Firstly, machine learning systems require large amounts of data for training [101] and are hard to interpret [167]. When analysing these systems, it is unclear which characteristics distinguish synthesised speech from bona fide. Therefore, knowing what humans use to identify deepfakes could provide a better understanding of how black-box machine learning systems work.

Secondly, focusing on automated biometric authentication does not quantify the threat of other potential criminal applications of speech deepfakes. Multiple studies deem other uses of speech deepfakes as more concerning, such as defrauding people through voice spoofs [161, 162].

We seek to address these two questions by measuring how well humans distinguish bona fide speech from synthesised speech. We ran an online experiment where individuals listened to bona fide and fake audio clips and attempted to differentiate between them.

We randomly assigned the participants to two conditions. In the first condition, we presented participants with one audio clip at a time and asked them to decide if the clip was fake. In the second condition, we presented participants with audio clip pairs containing the same speech (one bona fide and one synthesised) and asked them to identify the synthesised audio.

We ran the experiment in English and Mandarin to understand if listeners used language-specific attributes to detect deepfakes and to observe if deepfake detection is more manageable in one language than another. Finally, we incorporated randomised interventions to evaluate whether familiarising participants with examples of speech deepfakes boosts detection performance.

Our results suggest the listeners had limited detection capabilities, and performance is similar between languages. Additionally, familiarising participants improved performance but only to a small extent.

5.2 Background on deepfakes

Deepfakes are synthetic media produced in the likeness of a person. They fall into the field of generative artificial intelligence (AI). Generative AI algorithms learn patterns and characteristics to create synthetic content similar to the original data. Deepfakes specifically refer to the outputs of generative AI that resemble humans and their actions. Deepfake media occur in different modalities:

1. **Images**: This modality contains static faces generated using varying techniques. These techniques include:

    - **Generation from scratch**: A generative adversarial network [101] or diffusion model [168] synthesises a fictional identity.

    - **Morphing**: Blending similar-looking faces to produce an identity containing the characteristics of the sources [169].

    - **Swaps**: A source face replaces the target in a different image [170].

2. **Video**: This modality features individuals performing actions. Currently, the techniques used to synthesise videos are similar to those used in images. Image synthesis techniques are applied at a frame level and stitched together to form a video.

3. **Speech**: This modality conveys information in a manner that sounds like a genuine person's voice. Although audio can refer to general sound synthesis, the terms "au-

> dio", "speech", and "voice" deepfakes are used interchangeably in academic literature. We refer to them as "speech deepfakes" for consistency.

In addition, deepfakes are either produced in the likeness of a known identity (**targeted**) or do not resemble a familiar identity (**untargeted**). For example, we can categorise video deepfakes of politicians as targeted. Targeted deepfakes are often referred to as "spoofs" in the literature. However, spoofs are a broader category that includes non-deep learning methods to imitate individuals [171]. Examples include dubbing by voice actors [172] and replay attacks, where the adversary uses a recording of the targeted individual [173]. Conversely, a generic face created from scratch and not conditioned to resemble a specific individual is untargeted. The deepfake detection literature commonly refers to legitimate content of humans as "bona fide". Zhang (2022) [174] contains further information on deepfake terminology.

5.2.1 Synthesising speech

Generative models are often used to synthesise speech. Speech synthesisers which use generative models follow a common framework:

1. **Data collection**: Several audio recordings of the speaker are collected.
2. **Pre-processing**: The audio recordings are converted into alternative formats to make it easier for the generative model to work with them.
3. **Training**: Processed audio recordings are fed to the generative model to learn the patterns and characteristics of the data. The trained model is often called a vocoder.

The frameworks often include text-to-speech (TTS) modules to make it easier to generate speech. The generative model also sees text transcriptions corresponding to the audio recordings in this setting. We depict a visualisation of this framework in Figure 5.1.

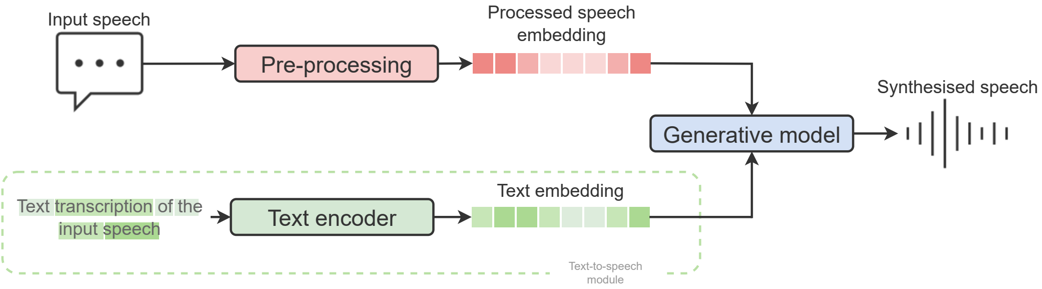


Figure 5.1: Diagram of a typical generative speech synthesis model.

5.2.2 Related work

Most deepfake detection studies which examine human performance use visual media. When faced with deepfake content of politicians, participants rely on contextual knowledge in the form of political literacy to identify spoofs [172, 175].

Removing such background knowledge makes the detection task more difficult. In the context of images, multiple studies show humans do not perform much better than chance [176, 177]. There is no improvement when evaluating videos either [9, 178, 179]. Moreover, these studies suggest humans are overconfident in their deepfake detection abilities [9].

Several of the above studies examine if interventions can boost detection performance. However, the effectiveness of these interventions is debatable. Bray et al. [176] familiarised participants by showing examples of deepfakes before the main task. The authors also drew participants' attention to errors often present in bogus images. Although these interventions improved deepfake detection performance, they also increased overall scepticism as a higher proportion of bona fide images were falsely classified. One could also note that pointing out errors biases the participants and prevents them from independently identifying the tell-tale characteristics of deepfakes. Köbis et al. [9] presented interventions by informing participants about the impact of deepfakes and rewarding correct guesses. Neither intervention led to improved performance.

In contrast, other authors found interventions derived from machine learning model outputs improve detection. Tahir et al. [179] produced educational material containing indicators of bogus images with the assistance of machine learning interpretability tools. The authors found detection performance improved compared to the initial control group. However, a recent study [180] contests the reliability of these tools, as the authors show it is possible to manipulate the output visualisations. Groh et al. [178] allowed participants to amend their choices after viewing the predictions of a machine learning model. This form of cooperation improved results significantly.

Fewer studies examine how well humans can detect speech deepfakes. Watson et al. [181] presented eight clips to college students and asked them to decide whether the clips were real or fake. They found that shorter clips were easier to identify. However, the sample size of their study was small and skewed towards a younger, college-educated demographic. The ASVspoof challenge organisers ran an experiment with a larger sample size [182].

They asked 1,145 participants to imagine they worked in a call centre and decide whether the incoming calls were spoken by humans or by an AI. However, the experiment was limited to the speaker verification setting.

Müller et al. [183] ran a game where 378 participants competed against a machine learning model to decide if an audio clip was fake. Similar to Groh et al. [178], they found that feedback from the machine learning model improved human performance. In their experiment, Müller et al. [183] found that the difference between human and AI accuracy was about 10%. However, their study only used English-language clips, only presented one audio clip to participants at a time, and did not collect information about participant confidence.

We summarise the relevant literature in Table 5.1. We note that Barari et al. [172] mention fake speech stimuli in their analysis. However, they used actors to create the speech instead of generative AI. Therefore we excluded this from our analysis.

Table 5.1: Summary of related literature measuring human capabilities to detect deepfakes.

Modality	Year	Author	Deepfake stimuli
Image	2021	Nightingale & Farid [177]	Faces generated using StyleGAN2 [184]
	2023	Bray et al. [176]	Faces generated using StyleGAN2
Video	2021	Barari et al. [172]	Face-swap videos of politicians
	2021	Groh et al. [178]	Face-swap videos from the Deepfake Detection Challenge dataset [185]
	2021	Köbis et al. [9]	Face-swap videos from the Deepfake Detection Challenge dataset
	2021	Tahir et al. [179]	Face-swap videos from Celeb-DF [186], FaceForensics++ [187] and DeepFaceLab [188]
	2022	Appel & Prietzel [175]	Face-swap videos of politicians
Speech	2020	Wang et al. [182]	Spoofed utterances generated from TTS and voice conversion systems used in ASVspoof2019
	2021	Watson et al. [181]	Audio clips generated using MelGAN [189]
	2022	Müller et al. [183]	Spoofed utterances from the ASVspoof2019 dataset [182]

5.3 Method

Our research questions were as follows:

1. How well can humans detect speech deepfakes?
2. Are there differences in detection capabilities depending on the language?
3. Do interventions in the form of examples and added context improve detection performance?

Through this setup, we could quantify the threat of speech deepfakes when humans interact with them. We anticipated these results would serve as a baseline for automated detection performance and inform the development of better automated detectors.

5.3.1 Stimuli

We introduce the bona fide and deepfake stimuli used in the experiments.

5.3.1.1 Bona fide stimuli

We collected bona fide stimuli from two publicly available datasets. Both datasets consist of one female speaker reading generic sentences. The datasets also include text transcriptions of the audio. We chose such datasets to prevent participants from using external cues for the detection task.

We used LJSpeech [190] as the English dataset. The dataset consists of a speaker reading passages from seven non-fiction books, varying between one and ten seconds in length.

We used the Chinese Standard Mandarin Speech Corpus (CSMSC) [191] as the Mandarin dataset. The corpus used in the dataset aims to cover Mandarin tones and prosody as comprehensively as possible.

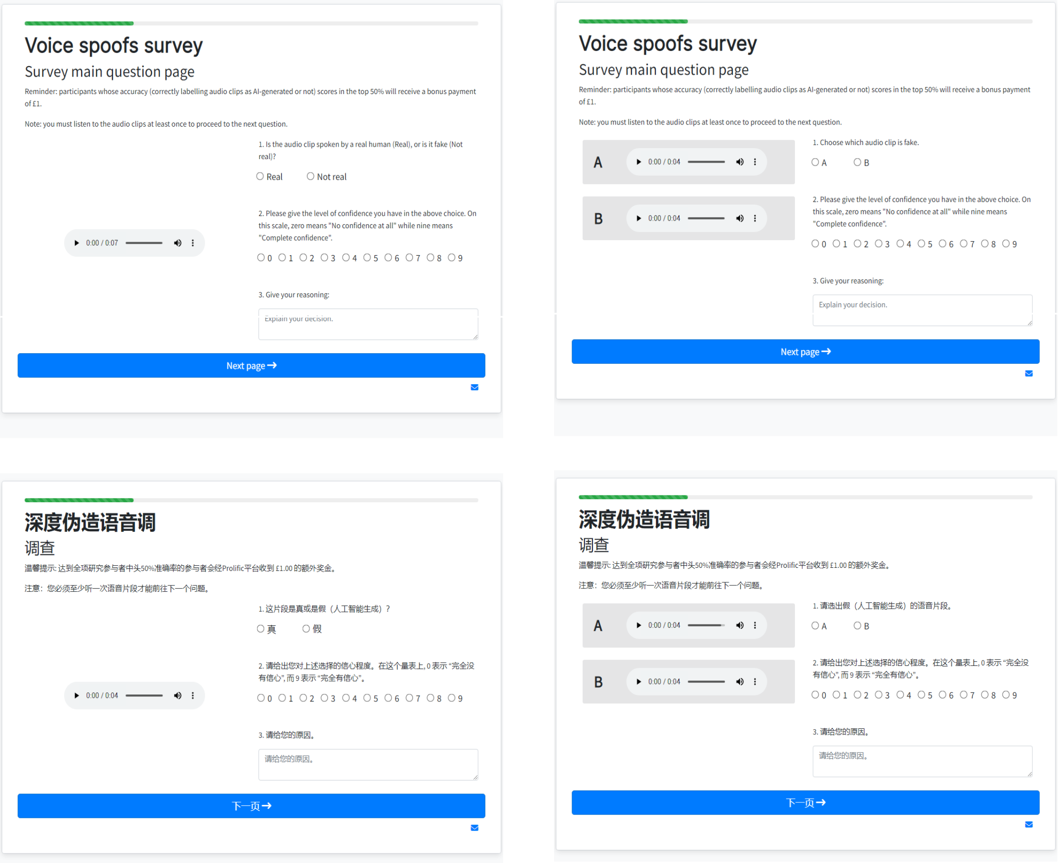
5.3.1.2 Deepfake stimuli

To create the deepfake stimuli, we used publicly available TTS models trained on the two datasets [192]. In particular, we chose pre-trained VITS models [193]. VITS is an end-to-end TTS model which combines the data pre-processing and vocoder into a single framework.

We randomly selected 50 sentences from the validation split of the two datasets to create the deepfakes. We used the same sentences for our bona fide stimuli. Therefore, we had 100 clips in total.

5.3.2 Procedure

The setup for the English and Mandarin experiments was identical. We randomly assigned participants to two configurations: unary and binary. In both configurations, we asked participants to rate the confidence of their choice on a ten-point Likert scale and provide freeform text justifications. Participants were allowed to listen to the clips as often as they liked. We did not give feedback to the participants to inform them if their choices were correct. Compared to the setups described in Groh et al. and Müller et al. [178, 183], the lack of feedback creates a more realistic scenario. When encountering speech deepfakes in the wild (for example, through fraudulent calls), humans do not know that the voices are fake. We include screenshots of the two configurations in Figure 5.2.



Voice spoofs survey

Survey main question page

Reminder: participants whose accuracy (correctly labelling audio clips as AI-generated or not) scores in the top 50% will receive a bonus payment of £1.

Note: you must listen to the audio clips at least once to proceed to the next question.

1. Is the audio clip spoken by a real human (Real), or is it fake (Not real)?

○ Real ○ Not real

2. Please give the level of confidence you have in the above choice. On this scale, zero means "No confidence at all" while nine means "Complete confidence".

○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9

3. Give your reasoning:

Explain your decision.

Voice spoofs survey

Survey main question page

Reminder: participants whose accuracy (correctly labelling audio clips as AI-generated or not) scores in the top 50% will receive a bonus payment of £1.

Note: you must listen to the audio clips at least once to proceed to the next question.

1. Choose which audio clip is fake.

○ A ○ B

2. Please give the level of confidence you have in the above choice. On this scale, zero means "No confidence at all" while nine means "Complete confidence".

○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9

3. Give your reasoning:

Explain your decision.

深度伪造语音调

调查

温馨提示: 达到全项研究参与者中头50%准确率的参与者会经Prolific平台收到 £1.00 的额外奖金。

注意：您必须至少听一次语音片段才能前往下一个问题。

1. 这片段是真或是假（人工智能生成）？

○ 真 ○ 假

2. 请给出您对上述选择的信心程度。在这个量表上, 0 表示 “完全没有信心”, 而 9 表示 “完全有信心”。

○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9

3. 请给您的原因。

请给您的原因。

深度伪造语音调

调查

温馨提示: 达到全项研究参与者中头50%准确率的参与者会经Prolific平台收到 £1.00 的额外奖金。

注意：您必须至少听一次语音片段才能前往下一个问题。

1. 请选出假（人工智能生成）的语音片段。

○ A ○ B

2. 请给出您对上述选择的信心程度。在这个量表上, 0 表示 “完全没有信心”, 而 9 表示 “完全有信心”。

○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9

3. 请给您的原因。

请给您的原因。

Figure 5.2: Screenshots of the task interface.

5.3.2.1 Unary configuration

We presented twenty randomly chosen distinct clips to each participant, each on separate pages. Participants listened to approximately an equal number of bona fide and synthesised clips, but we did not inform them about the proportion. We tasked the participants with deciding whether the clip they heard was real or fake.

5.3.2.2 Binary configuration

We presented twenty randomly chosen clip pairs (labelled "A" and "B") comprising the same spoken sentence. Each pair contained a clip uttered by the human speaker and a clip produced by VITS. We randomised the order of the fake and real clips and asked the participants to decide which clip was fake. We included this scenario to see if contextual information helped detection.

5.3.2.3 Familiarisation treatment

In addition to the two configurations, we randomly assigned half of the participants to a familiarisation treatment group. We included the treatment to verify the existing literature and understand if humans could be trained to detect deepfakes like a machine learning model. We showed participants in the treatment group five deepfake utterances before commencing the main detection task. We informed the participants that these examples were synthesised and allowed them to listen to the clips multiple times. These clips were distinct from the stimuli used in the main task.

For participants in the control group, we gave them a filler task. In this task, we asked participants to list potential applications of synthesised speech and to provide their opinion about whether synthesised audio will positively or negatively impact society.

5.3.3 Participants

We recruited participants via the Prolific platform. We filtered for participants fluent in English and Mandarin, as fluency affects detection performance [183]. We paid participants at a rate of £7.25 per hour. To encourage more thoughtful responses, we informed participants they could receive a £1.00 bonus if their detection scores were in the top 50%. Overall, we recruited 529 participants. The mean age was 28.9 years old, and 50.6% identified as male. Table 5.2 contains a more detailed breakdown of the demographics by treatment group.

Table 5.2: Number of participants by group.

Group	English			Mandarin		
	n	*Age (SD)*	*Male (%)*	*n*	*Age (SD)*	*Male (%)*
Unary no familiarisation	76	26.8 (8.1)	55.2	65	31.0 (10.4)	44.6
Unary familiarisation	65	26.7 (7.3)	56.9	54	31.4 (8.7)	44.4
Binary no familiarisation	60	27.5 (7.2)	53.3	70	31.8 (9.0)	48.8
Binary familiarisation	80	27.4 (7.3)	57.5	59	29.1 (8.5)	39.6
Overall	281	27.1 (7.5)	55.8	248	30.9 (9.2)	44.5

5.3.4 Ethics statement

The study was reviewed and exempted by the Department of Security and Crime Science's ethics board at University College London. All participants were notified about the purpose of the study and were over the age of 18. Before participating, the participants were asked to tick a series of checkboxes to provide informed written consent.

5.3.5 Benchmarking against automated deepfake detectors

To compare the performance of the human participants to automated methods, we trained two artificial neural networks which specialised in detecting speech deepfakes. Both networks used an LFCC-LCNN architecture [194], which converts raw audio waveforms into two-dimensional representations. The networks are trained on labelled bona fide and deepfake samples. The ASVspoof 2021 challenge used LFCC-LCNNs as baseline models for spoof detection [165]. Hence, they are a reasonable benchmark for our experiments. The article summarising the ASVspoof 2021 competition contains more detail about the top-performing speech deepfake detection architectures [165].

We used two versions for each language:

1. In-domain: We trained the networks using the training split of LJSpeech and CSMSC as bona fide samples and created deepfakes by passing the sentences of the training splits through VITS.

2. Out-of-domain: We trained the Mandarin network with FAD [195], another Mandarin-language dataset. We used the pre-trained ASVspoof network [196] for English-language evaluation.

We introduce the out-of-domain variant for a fairer comparison with human performance, as artificial neural network performances can decline with slight changes in the audio clips (such as changes to the speaker identity or environment). In addition, it is unlikely that the participants in our study recognise the identities in the LJSpeech and CSMSC datasets.

5.4 Results

We present overall detection results and analyse the effects of different interventions.

5.4.1 Overall performance

Figure 5.3 summarises human performance across all the different groups. We provide breakdowns of the classification choices in Tables 5.3 and 5.4, which aggregate the English and Mandarin results. We completed the analysis using the SciPy [197] and statsmodels [198] Python packages. For further details, Appendix C.3.1 contains results per stimulus.

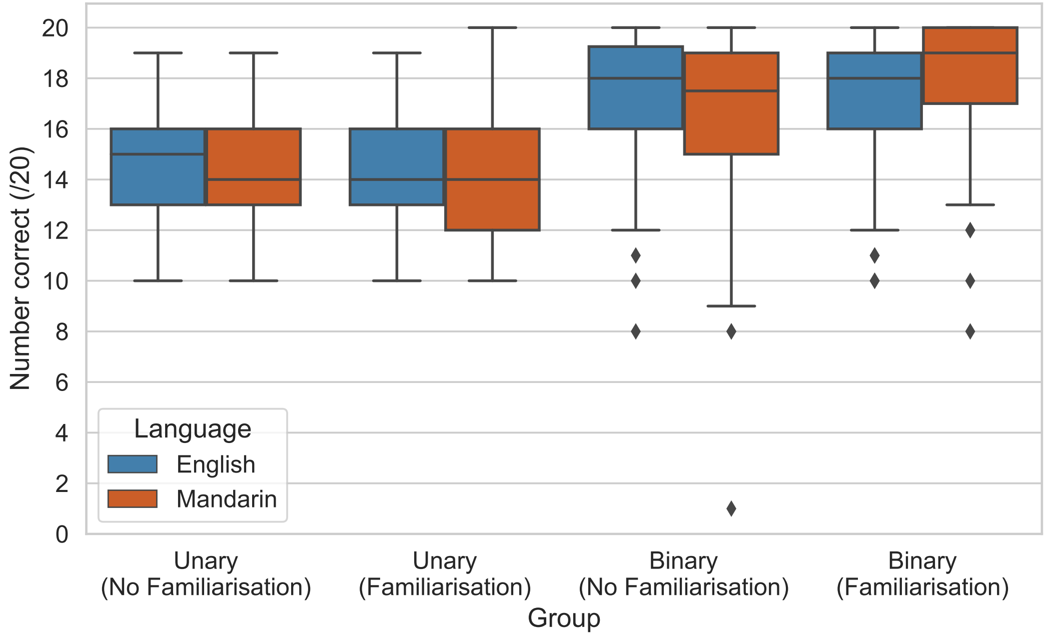


Figure 5.3: Box plot summarising human performance across the different groups.

Participants made the correct classifications 70.35% of the time in the unary scenario. They were better at identifying deepfakes (73% accuracy). In comparison, participants correctly identified bona fide examples 67.78% of the time. We speculate the high number of misclassified bona fide samples is partly due to increased scepticism, as participants were aware of the presence of deepfakes through the task briefing. This behaviour aligns with

Table 5.3: Confusion matrix for the unary group responses.

		Predicted class	
		Real (2,442)	*Fake (2,678)*
True class	*Real (2,598)*	1,761	837
	Fake (2,522)	681	1,841

n = 5,120.
Overall accuracy = 70.35%.
Reals correctly identified = 67.78%.
Fakes correctly identified = 73.0%.

Table 5.4: Confusion matrix for the binary group responses.

		Predicted class	
		Real	*Fake*
True class	*Real*	-	-
	Fake (5,380)	775	4,605

True *real* class labels are not defined in this scenario as participants were asked to choose the fake clip every time.
Overall accuracy is equivalent to fakes correctly identified = 85.59%.

observations in Bray et al. [176].

Performance improved under the binary scenario. Participants correctly recognised the deepfake audio in 85.59% of trials. However, the binary setup represents an unrealistic scenario. Even if the identity of a speaker is known, reference utterances containing the same speech as the test clip we would like to evaluate are unlikely to be available.

5.4.2 Measuring the effects of interventions

We follow a similar approach to Groh et al. [178] to disentangle the effects of each intervention on performance. We transformed the correct/incorrect results into continuous values by weighting the decision of each participant with their provided confidence scores.

The ten-point confidence scale participants completed serves as the mapping function. The lowest score of 0 signals that the participant's choice is a guess, so their confidence in making the right decision corresponds to 50%. In contrast, the highest score of 9 corresponds to 100% belief.

The resulting transformed scores depended on whether the participants made the correct classification. For example, if the participant rated their confidence as 7, this maps to a belief of 88%. If they make the right decision, the adjusted score is 0.88. Conversely, if they make the wrong decision, we subtract the value from 1, resulting in an adjusted score of 0.12.

The revised scores also enable fairer comparisons with the automated deepfake detectors, which output scores between 0 and 1 when evaluating examples. We refer to the revised scores as accuracy scores for the remainder of the text. We also rescale the scores to percentages.

After transforming the results, we analysed the effects of different interventions on the accuracy scores of participants on each audio clip using linear regression. In addition to language, familiarisation and binary intervention, we analysed the impact of the clip duration. Table 5.5 outlines the results at the overall, unary and binary levels.

Table 5.5: Linear regression results of interventions on confidence-scaled accuracy.

Independent variable	**Dependent variable: Confidence-scaled accuracy**		
	All (SD)	*Unary (SD)*	*Binary (SD)*
Constant	43.742*** (1.897)	46.394*** (2.791)	71.217*** (2.530)
Mandarin[a]	1.790 (1.404)	1.477 (1.882)	2.152 (2.118)
Familiarisation	3.840*** (1.191)	3.758** (1.571)	3.854** (1.802)
Clip length	0.797*** (0.209)	0.375 (0.358)	1.168*** (0.230)
Binary intervention	29.830*** (1.186)	-	-
Observations	10,500	5,120	5,380
R^2	0.165	0.003	0.010
Adjusted R^2	0.165	0.002	0.009
F-Statistic	171.102***	2.455*	11.794***

[a] Dummy variable indicating which language was used in the task. 1 = Mandarin, 0 = English.
*$p < 0.1$. **$p < 0.05$. ***$p < 0.01$.

5.4.2.1 Reference audio helps with deepfake detection

The linear regression results indicate the improvement gained from the binary scenario is statistically significant ($p < 0.001$). Consequently, the results suggest contextual information via reference audio is beneficial for uncovering quirks in synthesised speech.

5.4.2.2 Training humans to detect deepfakes only helps slightly

The familiarisation treatment increases detection accuracy by 3.84% on average ($p = 0.001$). This effect is present in both the unary and binary regression results, improving accuracy by 3.76% ($p = 0.017$) and 3.85% ($p = 0.032$), respectively. However, the familiarisation treatment only boosts accuracy to a level which is slightly above chance in the unary setting, ceteris paribus.

5.4.2.3 It is equally challenging to detect deepfakes in Mandarin and English

Figure 5.3 shows that performance in English and Mandarin is comparable across the different treatment groups. This observation is supported by Table 5.5, which shows Mandarin-speaking participants only outperform their English counterparts by 1.79%, and this effect is not statistically significant ($p = 0.202$).

5.4.2.4 Shorter speech deepfakes are not easier to identify

As our stimuli varied from two to eleven seconds, we included clip length in the regression to verify whether it is easier to discriminate shorter clips. Our results suggest clip length has a negligible impact on accuracy, improving performance by only 0.80% for each additional second. Our scatter plot (Figure 5.4) supports this and shows no relationship between the two variables. These findings conflict with Watson et al. [181], who suggest it is easier to identify shorter deepfakes.

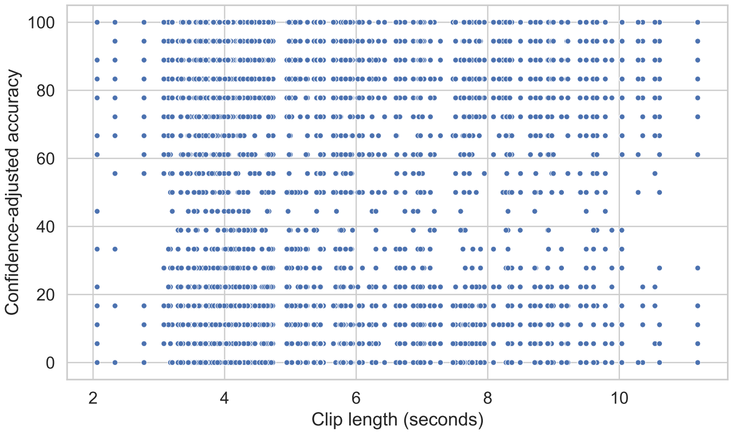


Figure 5.4: Scatter plot showing the relationship between clip length and confidence-scaled accuracy.

5.4.3 Analysing performance against time

In addition to analysing the treatment effects, we examine whether the hypothesis of spending more time on the task improves performance.

5.4.3.1 Listening to the clips more frequently does not aid detection

We recorded the number of times participants clicked on each audio clip and compared the values to accuracy. As shown in Figure 5.5, there is no relationship between the two variables ($\rho = -0.05$, $p < 0.001$).

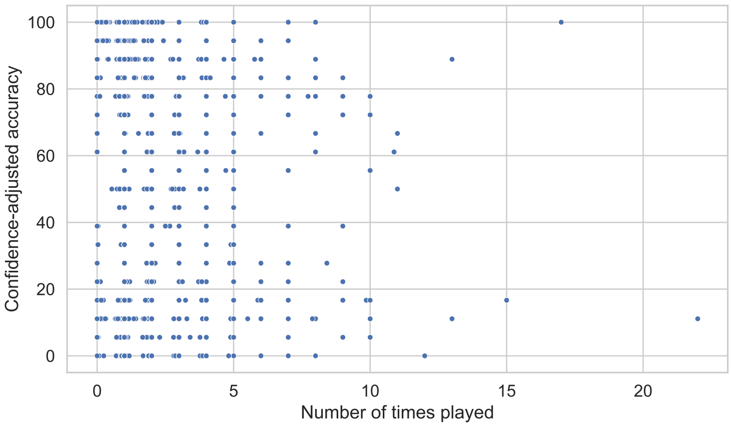


Figure 5.5: Scatter plot showing the relationship between the number of times played and confidence-scaled accuracy.

5.4.3.2 Spending more time on the task also does not affect performance

Similar to the above analysis, we compared the time taken to complete the entire task to the total number of clips correctly identified. Figure 5.6 only suggests a weak relationship between the two variables ($\rho = 0.10$, $p = 0.018$), suggesting investing more time to complete the task does not improve performance.

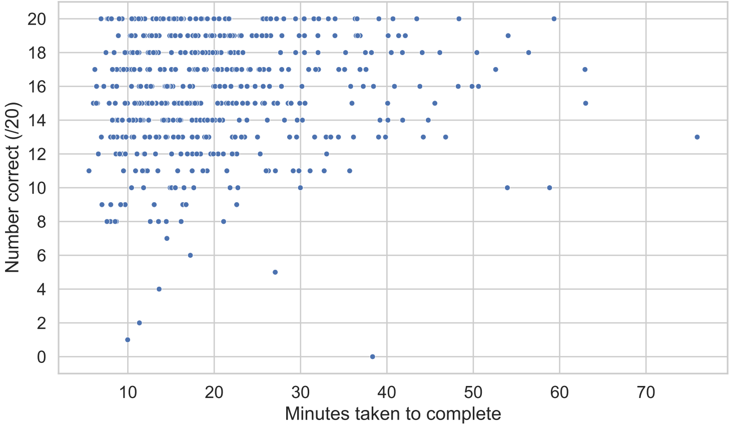


Figure 5.6: Scatter plot showing the relationship between minutes taken to complete and correctness scores.

5.4.3.3 Participants do not get better throughout the task without explicit feedback

To understand whether participants improved as they saw more examples and progressed further in the task, we calculated the number of correct responses per question number. If so, we would expect more correct answers in question twenty compared to question one. Figure 5.7 illustrates the resulting histogram. The histogram shows performance is relatively stable across the questions. This observation indicates participants do not improve throughout the task unless they have explicit feedback, as examined by Groh et al. [178] and Müller et al. [183]. We quantitatively verified the result by conducting a one-way chi-squared hypothesis test against the uniform distribution, which was not statistically significant ($\chi^2 = 6.19$, $p = 0.997$).

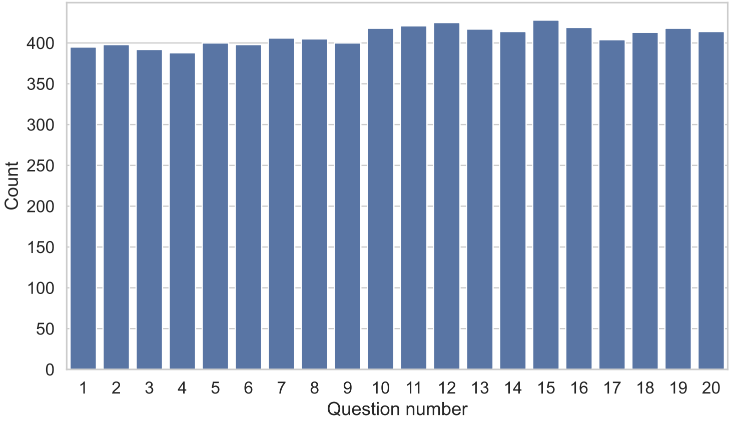


Figure 5.7: Histogram of correct responses across question number.

5.4.4 Comparing human performance to automated detectors

The following section compares human performance to automated deepfake detectors. For comparability, we use commonly reported performance metrics found in machine learning literature.

- **Receiver operating characteristic (ROC)**: These plots represent discriminatory ability. They compare true positive rates against false positive rates at different thresholds.
- **The area under the receiver operating characteristic (AUROC)**: This score summarises ROCs into a single value. 50% AUROC indicates predictions are guesses, whereas 100% AUROC means perfect discrimination between bona fides and deepfakes in all trials.
- **Equal error rate (EER)**: This describes the point on ROCs where the true positive and false positive rates are equal.

Figure 5.8 displays the AUROC and EER scores. We include only the unary scenario in this analysis as the inference setup between humans and automated detectors is more comparable. Both evaluate one clip at a time. We aggregated the English and Mandarin results as we observed similar results.

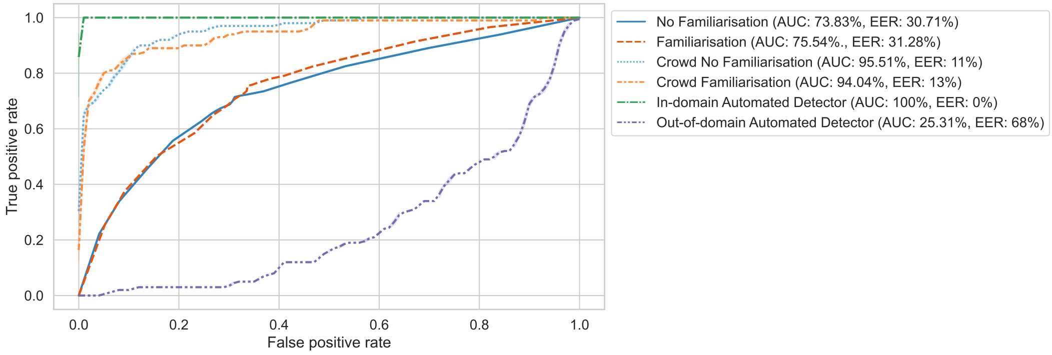


Figure 5.8: Receiver operator curves under the unary scenario.

5.4.4.1 Human performance is less sensitive to unknown conditions compared to automated detectors

The no familiarisation (AUROC = 73.83%) and familiarisation curves (AUROC = 75.54%) confirm humans performed better than chance. The curves also support the linear regression result. Showing participants examples of deepfakes only had a minute impact on performance. However, performance was quite unreliable: on average, humans incorrectly classified clips a quarter of the time.

Humans underperformed the in-domain automated detectors, which had perfect discrimination ability (AUROC = 100% for both languages). However, the out-of-domain detectors often incorrectly classified bona fides as deepfakes (AUROC = 25.31%). Based on this behaviour, humans are more robust to unknown factors, such as speaker identity.

5.4.4.2 Crowd-sourced speech deepfake detection is comparable to the top-performing automated detectors

Per Groh et al. [178], we averaged the accuracy scores of each participant per clip to calculate the crowd-sourced responses. Like the results observed with video stimuli [178], crowd-sourced performance is on par with the in-domain detector. However, the benefit of familiarising participants dissipates when averaging responses. The crowd-sourced no familiarisation and crowd-sourced familiarisation AUROC scores are similar at 95.51% and 94.04%, respectively.

5.4.5 Freeform text analysis

To understand how participants assessed the genuineness of audio clips, we analysed their freeform text responses. We grouped responses by language, clip authenticity, and whether participants made the correct choice. We then created word clouds using tf-idf weightings. Tf-idf measures the importance of a word within a document compared to a collection of documents to account for frequently appearing words [199]. Figures 5.9 and 5.10 show the English and Mandarin word clouds.

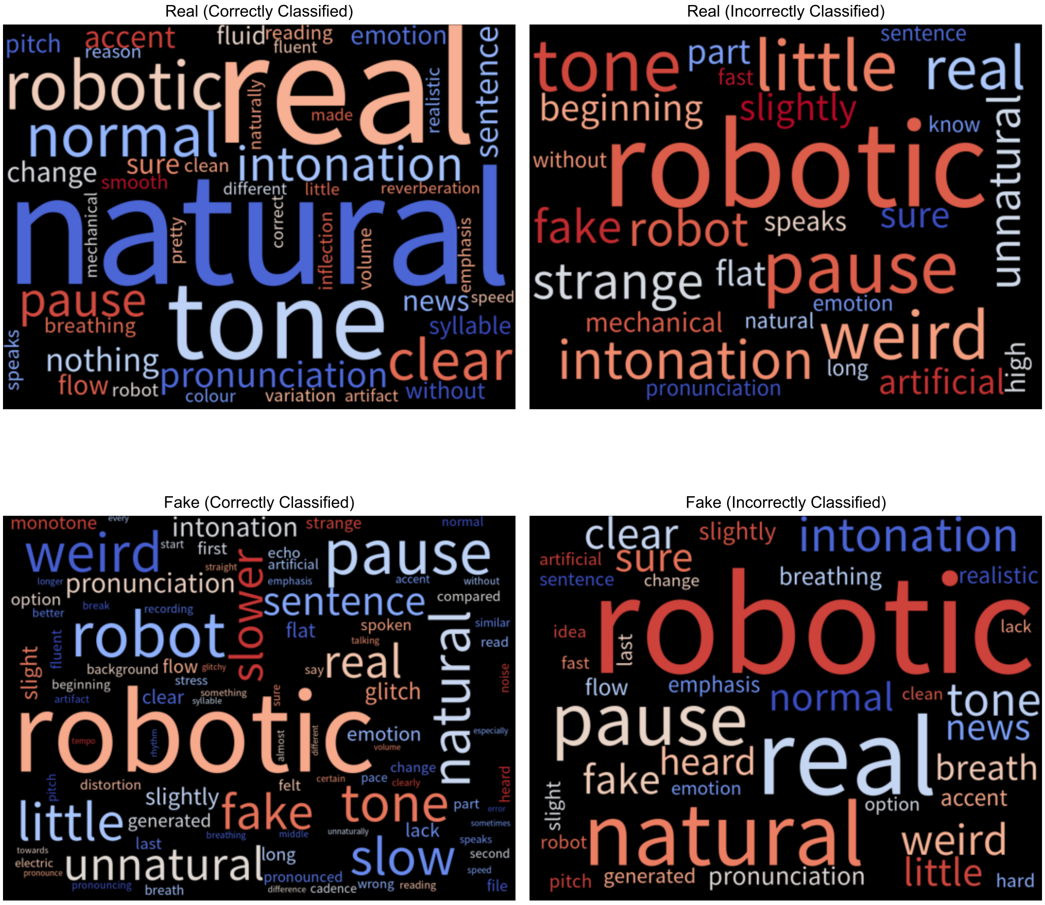


Figure 5.9: Word clouds containing justifications for the English-language clips.

Participants referred to the same characteristics regardless of whether they made the correct decisions. For example, in Figure 5.9, participants who correctly classified bona fide utterances as legitimate (in the top left of Figure 5.9) mentioned pauses, tone and intonation. However, participants who incorrectly categorised bona fide utterances as fake (top right of Figure 5.9) also referred to these attributes. We compared responses by the actual label of the clips and whether participants made the correct response. We did not find substantial differences between these segments. Therefore, automated detectors that incorporate these human characteristics would produce limited improvements. We observed this activity in both English and Mandarin. They tended to rely on intuition to make classifications, referring to naturalness (自然) and robotic (机械) sounds. Beyond intuition, English and Mandarin participants also commonly referenced pauses (停顿), intonation (语调), pronunciation (发音), and speed (速度).

Regarding differences between languages, there were more references to breathing among

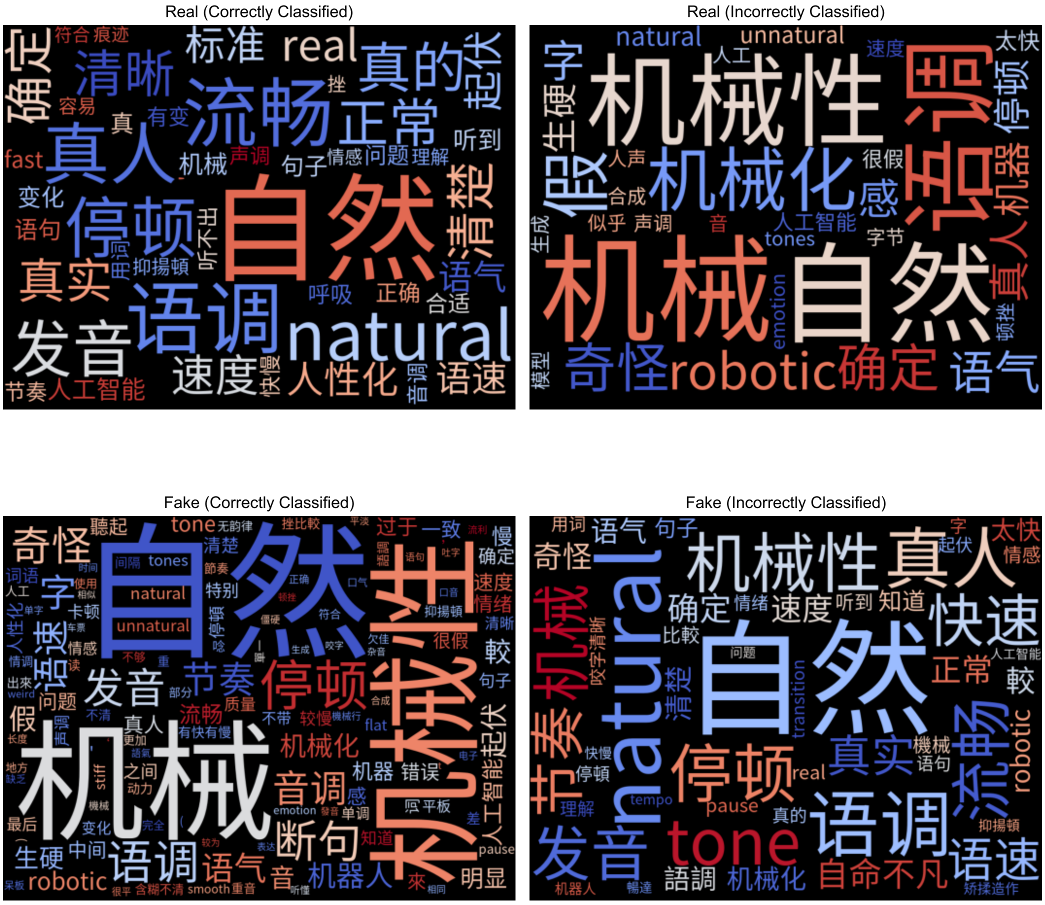


Figure 5.10: Word clouds containing justifications for the Mandarin-language clips. Note participants for the Mandarin tasks provided justifications in both Mandarin and English.

the English-speaking participants. In contrast, Mandarin-speaking participants mentioned the cadence of the speaker (节奏), pacing between words (断句), and fluency (流畅). This result may be due to differences in timing properties between the two languages. English is stress-timed, while Mandarin is syllable-timed [200].

5.5 Conclusion

We conclude by outlining the work’s limitations and main contributions.

5.5.1 Limitations

Although our setup enabled comparison with automated detectors, it does not necessarily reflect more realistic scenarios where a listener may encounter speech deepfakes.

Firstly, the balance of deepfakes we presented in our experiment does not reflect the proportion that occurs in the wild. Participants were equally likely to encounter deepfakes as bona fides in the task. However, AI-generated content (including the use of deepfakes for nefarious purposes) is still rare for now. In addition, we could expect participants to be much more attentive to the occurrence of deepfakes as we informed them about the nature of the task.

Moreover, we minimised contextual information in our stimuli. For example, we do not examine situations where the contextual knowledge of a listener (such as awareness of the speaker's identity, emotional status, the number of parties in a conversation, or political affiliations) may have informed their decisions. These aspects may be relevant to typical use cases where speech deepfakes may arise, such as false news propagation [161]. Future work could look at exploring how these characteristics influence detection.

Additionally, we asked participants in both languages to listen to utterances purporting to originate from a single female speaker. Given that the age and gender of speakers influence speech perception [201, 202], future work could consider how varying speaker identity affects deepfake detection performance.

To generate our deepfake stimuli, we used an older approach which is not necessarily illustrative of the state-of-the-art speech synthesis algorithms. Although our results indicate how well humans can detect speech deepfakes generated with limited computational resources, they may not faithfully reflect performance under the most current conditions.

5.5.2 Summary

Humans can detect speech deepfakes, but not consistently. They tend to rely on naturalness to identify deepfakes regardless of language. As speech synthesis algorithms improve and become more natural, it will become more difficult for humans to catch speech deepfakes.

Although there are some differences in the features that English and Mandarin speakers use to detect deepfakes, the two groups share many similarities. Therefore, the threat potential of speech deepfakes is consistent despite the language involved.

It will be easier for adversaries to generate more deepfakes as the computational barrier for synthesising data lowers. More deepfakes in the wild will have a knock-on effect. Adversaries will have more opportunities to scale their operations, particularly for disinformation such as impersonations and spear phishing [162].

Ultimately, the battle between deepfake creation and detection is an arms race [203]. How can we defend against falling prey to deepfake trickery? Our binary scenario shows that comparing against reference audio is helpful if we know the speaker's identity. However, we do not always have this information.

Increasing awareness by showing people examples of deepfake audio has a limited effect, as demonstrated by our familiarisation results. Spending more time evaluating the clips does not seem to help either.

To summarise, attempting to improve human detection capabilities is unrealistic. We show that even in a controlled environment where the task is easier (participants are aware of the presence of speech deepfakes and the deepfakes are not created using state-of-the-art speech synthesisers), deepfake detection is not high. Our results suggest the need for automated detectors to mitigate a human listener's weaknesses. Automated detectors' performance on in-domain data indicates they can pick up on subtleties that humans cannot. However, we show they are brittle and fail to work when there are changes in the test audio's environmental conditions. Given the extent of human limitations and the increasing availability of computational resources for deploying detectors, research should focus on improving these detectors. We attempt to tackle this issue in the next chapter by analysing how deepfake detection performance varies with different representations.

In the meantime, crowd-sourcing is a reasonable mitigation. We confirm crowd performance is on par with the top-performing automated detectors and is not as brittle. Extending fact-checking tools to include audio evaluations is one way to protect against deepfake threats.

6 | Speech: automated deepfake detection

Existing automated deepfake detection approaches tend to use binary classifiers [204]. These classifiers learn by distinguishing between deepfake and bona fide speech. At test time, these classifiers work well when presented with clips similar to those seen during training. However, their performance degrades when there is a distribution shift in the test audio [14, 204, 205]. For example, changes in the speaker identity or added compression (such as transmission through telephone lines) can result in these shifts.

Improving the generalisability of deepfake detectors is an ongoing research topic in the speech-processing community. One common strategy uses embeddings from large pre-trained neural networks as input to binary classifiers [206, 207, 208]. Pre-trained networks transform deepfake and bona fide training samples into a different representation. The binary classifier then learns to differentiate bona fide and deepfake utterances in this new space. Pre-trained networks benefit from learning high-level features that transfer well to several tasks because they encounter varied datasets at the pre-training stage. However, this setup does not guarantee that the features will transfer well to all unseen anomalies.

Binary classifiers are prone to learning brittle decision boundaries [209]. As a result, they often yield unexpected results outside their decision space. Our experiments in §5.4.4 confirm this behaviour: performance on the OOD data was worse than chance. The binary classifier evaluated on OOD bona fide and spoof data achieved an AUROC of 25.31%.

An alternative strategy would be to make decisions on the pre-trained embeddings directly. This approach removes the computational expense of an additional classifier. Furthermore, we can use bona fide data to set up a one-class detector. Training a model with only bona fide data reduces the potential to overfit on particular deepfake synthesis methods.

Although one-class anomaly detection is effective in other modalities like image and text [113, 210], its usefulness for speech deepfake detection is underexplored. In the absence of labelled data, images and text have benefitted from pretext tasks that take advantage of intrinsic biases to learn benign representations. Intrinsic biases include translation invariance in images or fixed grammatical structures in text. Examples of useful pretext tasks that leverage these biases include denoising [82, 102], autoregressive modelling [31], and contrastive learning [40].

It is not straightforward to adapt these tasks to speech. Audio clips cannot undergo discretisation like text. Although contrastive learning is domain-neutral, it requires appropriate augmentation strategies to learn good representations [40]. It is unclear which augmentations preserve the discriminative features necessary for speech deepfake detection. Additionally, several components for effective speech deepfake detection are unknown, such as the most appropriate audio representation, how to encode knowledge of the bona fide distribution, and the choice of detector.

Instead of using bespoke pretext tasks, we can take advantage of the nature of bona fide data. Bona fide speech is widely available and can come from various sources, like different speakers. Therefore, we can use the metadata to construct a pretext task. Alternatively, we can treat bona fide data as one homogenous class and use a generic centre loss to learn a compact embedding [65, 66, 67].

We investigate whether one-class models can detect speech deepfakes and if they have the same shortcomings as their binary counterparts. We study multiple aspects of the one-class anomaly detection pipeline, including the choice of representation space, detector, and architecture.

Our results are as follows:

1. One-class anomaly detectors can detect deepfakes and outperform binary classifiers on unseen bona fide data.
2. Fine-tuning pre-trained models with bona fide data is the most beneficial. However, most of the performance gains come from the underlying pre-trained model.
3. Embeddings extracted from more generalised audio classification models are better for anomaly detection than embeddings from models trained specifically on speech.

4. The most effective embeddings occupy a low-dimensional subspace. We can measure this property by projecting data to a lower subspace with principal components analysis and measuring performance on a proxy task.

From these findings, we provide the following contributions:

1. Our experiments show that one-class anomaly detectors can detect speech deepfakes.
2. We show that one-class anomaly detectors can complement binary classifiers, as they are more robust to unseen bona fide data.
3. We provide practical insights on how to perform one-class deepfake detection. Key components include the choice of underlying pre-trained network, the fine-tuning objective, and the type of anomaly detector.
4. We diagnose conditions where anomaly detection should succeed by analysing the datasets in detail.

6.1 Background

We discuss different approaches for deepfake detection: binary classifiers, one-class learning for speech data, and one-class learning in other modalities.

6.1.1 Binary classifiers

Deepfake binary classifiers typically have two components: a frontend (which transforms the input features) and a backend (which makes the classification decisions).

Early iterations used hand-crafted time-frequency representations as the front end. These methods fed Mel frequency cepstral coefficients (MFCCs) [211] to shallow backend classifiers like Gaussian mixture models, support vector machines and probabilistic linear discriminant analyses [182]. Subsequent backends switched to CNNs to process time-frequency features more effectively. The consensus was that CNNs could identify more fine-grained patterns that differentiate deepfakes from bona fide utterances due to their ability to perform localised feature extraction [26].

However, time-frequency representations come at the cost of information loss, such as phase and temporal resolution. This information loss could remove important discriminative features. These representations also vary depending on hyperparameters like window size, shift length and dimensionality, leading to inconsistencies across implementations. RawNet [212] and RawNet2 [213] attempt to resolve these issues by modifying the CNN architecture to permit raw waveforms as input, replacing the two-component system with an end-to-end approach. A variation of the end-to-end approach uses graph neural networks (GNNs) instead of CNNs [214]. These GNNs aim to surpass CNN performance by processing the time and frequency domains in separate neural network branches before combining them at a later processing stage.

More recent approaches using GNNs have reverted to separate frontend modules [215]. These frontend modules use features from self-supervised models designed for speech, including HuBERT [216] and Wav2vec 2.0 [217, 218].

HuBERT adapts the BERT language model to speech [82]. HuBERT predicts masked speech tokens instead of words. It learns to convert the raw waveforms into discrete units like the language tokens in BERT. The conversion process relies on k-means clustering. Once converted, a proportion of the units undergo masking. A transformer encoder in the HuBERT architecture predicts the values of the masked components during training.

Wav2vec 2.0 has a different training setup. Instead of k-means clustering, it discretises the waveforms using a quantisation module. Training also relies on contrastive predictive coding. Contrastive predictive coding involves predicting parts of the audio based on other segments of the utterance [219].

We provide a more detailed overview of the models in Appendix B.2.

6.1.2 One-class learning for speech

In recent audio deepfake detection literature, "one-class learning" refers to the work of Zhang et al. [220]. They propose a new loss function called "one-class softmax". This loss function encourages the model to learn a feature space that groups the bona fide embeddings into a tight space and pushes the deepfake embeddings away by a fixed margin. The anomaly scorer uses the model's classification confidences. Lower confidence scores signal more anomalous utterances. Their approach is a misnomer, as the training stage includes labelled deepfakes. It is a supervised contrastive learning method.

Ding et al. expand on the work of Zhang et al. [220] by introducing speaker attractor multi-centre one-class learning [221]. Ding et al. found that bona fide utterances often form multiple clusters in the embedding space, as data tends to comprise multiple speaker identities. As a result, grouping all bona fide data into one unit could result in misclassifications. They modify the bona fide class in the one-class softmax loss function to set multiple speaker identities as attractor classes. The attractors receive updates during optimisation to improve the positions of the speaker identities. The model continues to learn with a margin-based objective. However, this approach requires speaker identity labels in addition to labelled spoofs.

Pianese et al. take a more traditional approach to one-class learning [222]. They extract speaker embeddings using a pre-trained model to initialise attractor classes. Each attractor class has a different speaker identity. They use cosine similarity and squared Euclidean metrics between a test sample and the attractor classes to evaluate test audio. However, this method also requires knowledge of speaker identities.

Few works train models using only bona fide data without speaker labels. Alegre et al. [223] use local binary patterns [224] of speech cepstrograms to train OCSVMs, demonstrating the viability of actual one-class learning. Villalba et al. use one-class learning as part of an ensemble submission to the ASVspoof 2015 challenge [225]. They extract features from a multilayer perceptron (MLP) to train an OCSVM. However, the approach was not purely unsupervised. The MLP learnt to classify between bona fides and spoofs.

6.1.3 One-class learning for deepfake detection in other modalities

One-class deepfake detectors for modalities outside of speech fall under various subcategories:

Outlier exposure. Outlier exposure techniques introduce a set of proxy anomalies at the training stage [154]. These proxy anomalies are either OOD data from another dataset or are synthetically generated. The model learns to classify between the bona fide samples and the proxies. Shiohara and Yamasaki [226] adopt this approach to detect face swap deepfakes. They synthesise deepfakes by conducting self-swaps. Self-swaps are face swaps that use the same identity as the source image. As a result, the model learns about the synthesis method rather than irrelevant background information.

An analogous approach in speech would be using a voice converter to transform a voice back to the source identity. However, this outlier exposure method specialises in detecting samples from a particular synthesiser. Therefore, it is not guaranteed to generalise to other unseen synthesis methods. To overcome this problem, a model would need to train on self-swaps synthesised using several methods.

Reconstruction. Reconstruction methods assume that models trained on bona fide data cannot represent anomalies [1]. The anomaly score is typically the input-output reconstruction loss [71], although other methods use the difference between latent embeddings [46, 72]. Khalid and Woo [227] apply the reconstructive approach to facial deepfake detection using autoencoders, which are popular reconstruction models. They train a variational autoencoder on genuine faces and use the root mean squared error to score test images. However, in certain instances, autoencoders can represent anomalies [228]. Feng et al. [229] switch to to autoregressive models to identify audiovisual deepfakes. They use a contrastive learning framework to synchronise the audio and video content. They extract features from the contrastive model to train an autoregressive transformer. The authors assume the transformer cannot fit a distribution to deepfakes, so they flag sequences with low probabilities as anomalous. However, generative models also suffer from the anomaly generation issue. Other studies suggest deep generative models can assign higher anomaly scores to in-distribution data than out-of-distribution data [73, 230, 231].

Client matching. Instead of comparing reconstructed data against its input, other approaches compare input data with an exemplar identity. This is analogous to speaker verification in speech [222]. Cozzolino et al. [232] extract a compact identity embedding using a three-dimensional morphable model that captures shape, expression and appearance using principal components analysis. The model transforms a video clip into this three-dimensional representation at test time. An anomaly scorer compares this representation against the claimed identity.

In a subsequent study, Cozzolino et al. extended client matching to the audiovisual modality [233]. They use contrastive learning to align training audio and video clips. This step ensures embeddings belonging to the same identity across modalities lie close to each other. During evaluation, they extract intermediate embeddings from the contrastive model. The anomaly score is the similarity between the test embedding and the closest reference identity. Similar to the speech deepfake case, this approach requires client labels.

Self-supervision. Although autoencoders fall into this definition, newer methods use pretext tasks that do not reconstruct data. Haliassos et al. [234] and Shi et al. [235] train self-supervised models on bona fide data before fine-tuning supervised classifiers. Although not purely one-class approaches, these methods suggest that training on bona fide data can provide better generalisation.

Haliassos et al. [234] adapt a student-teacher approach [62]. The teacher model receives audio data. The student model receives video data but must match the teacher's output with a mean squared error objective. The supervised classifier has two learning objectives. It learns to predict what is bona fide or deepfake. It also learns to predict the embeddings of the teacher.

Shi et al. [235] pre-train a transformer with a masked image modelling objective [236]. Following this step, they mask all blocks and obtain predictions for each block. They compute a residual map using the difference between the input and the predictions. They then use the map as the input to a supervised classifier that learns to predict between bona fide and deepfake images.

Pre-trained embeddings. Self-supervision is computationally expensive as it needs large batch sizes [40]. Other works show that performing anomaly detection with pre-trained embeddings works well, which raises the question of whether additional fine-tuning on bona fide data is necessary. Many post-hoc approaches that use pre-trained embeddings extract features from the pre-logits layer and run shallow classifiers such as k-NN or Mahalanobis distance [16, 209]. Some studies build on the pre-trained embedding by fine-tuning with the bona fide data using a centre loss [67]. However, the centre loss is prone to representation collapse [66, 67]. There are strategies to avoid collapse, such as early stopping and regularisation [67, 237].

6.2 Method

Our research questions were as follows:

1. Can we use one-class anomaly detection to identify speech deepfakes? If so, what

representations work the best?

2. Given a fixed embedding, how do different anomaly detectors compare?

3. Do anomaly detectors perform better on unseen data distributions compared to their supervised counterparts?

Our experimental pipeline follows a set structure, as illustrated in Figure 6.1:

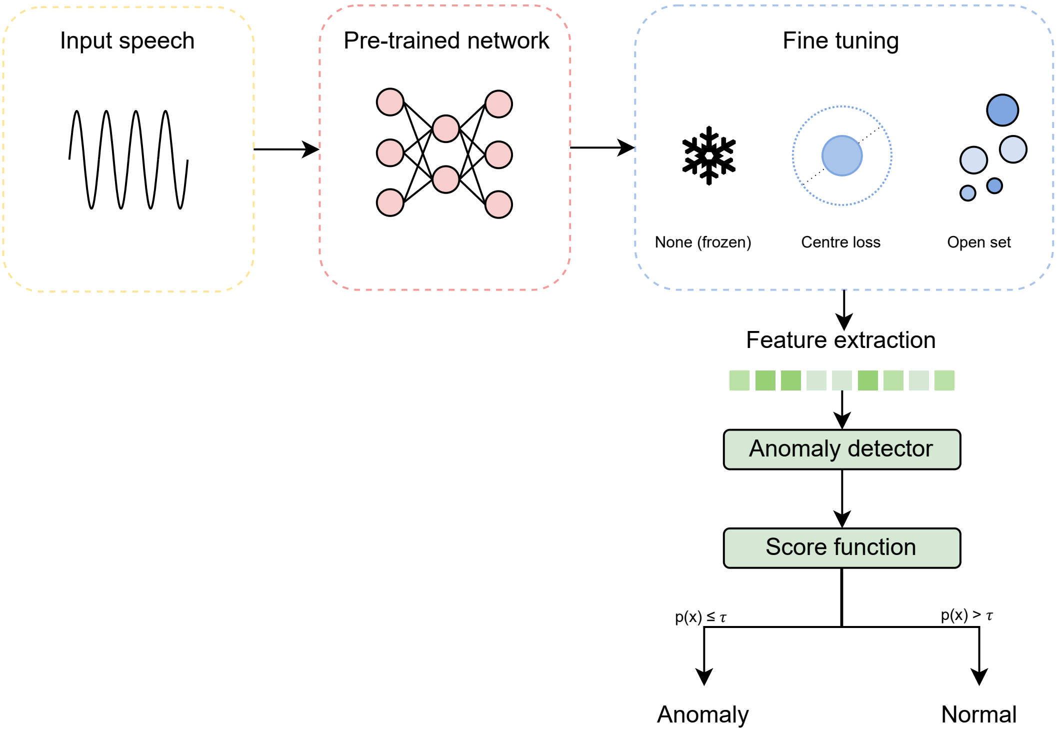


Figure 6.1: Diagram of the experimental pipeline.

Embedding extraction. We use the pre-trained neural network as is, or after fine-tuning. We pass the training data through the neural network and obtain a new embedding from the hidden layers. We use these embeddings to approximate the bona fide distribution.

Anomaly scoring: We pass the test data through the same neural network to transform it into a new representation space. We compare how similar it is to the approximated bona fide distribution using a shallow detector.

Our anomaly detection setup has the following advantages:

- **Minimal engineering**: Using neural networks as feature extractors means we do not need prior knowledge to create hand-crafted embeddings. We also do not need additional supervised classifiers as we perform anomaly detection directly on the embeddings.

- **Speaker agnostic**: We do not need to use labelled speaker identities to refine the anomaly detection system.
- **Only needs bona fide data**: We do not need to collect deepfakes samples to train the system, which should encourage better generalisation to unseen deepfake synthesis methods.
- **Architecture independent**: The extraction and shallow anomaly detection method can be applied to various architectures. We show this by analysing the effect of different pre-trained audio and speech transformers on detection capability.

We vary model architecture, training objective, anomaly detector, and training data to understand the importance of each component. We clarify these aspects in the following sections.

6.2.1 Datasets

We train and evaluate the anomaly detectors on publicly available English, Mandarin and Japanese deepfake datasets. We used ASVspoof [163, 204] to align with previous deepfake detection work. We chose additional Mandarin and Japanese datasets to ensure our results were applicable beyond English.

ASVspoof 2019. The ASVspoof challenge, initially created in 2015, aimed to encourage research on anti-spoofing [182]. It established an initial benchmark of spoofing countermeasures. ASVspoof 2019 was the third edition of the challenge [163] and was the first to introduce logical access (LA) attacks. LA attacks involve scenarios where an adversary seeks to break into systems remotely using deepfakes. We only use the LA subset of ASVspoof 2019, as the other partition uses presentation attacks. Presentation attacks (playback of recorded voices) are not in scope for this work as they are not products of deep learning models. The database uses the Voice Cloning Toolkit (VCTK) corpus [238], a multi-speaker English speech database, as the bona fide samples. The training split has 2,580 bona fide and 22,800 deepfake utterances. The test split has 5,370 bona fide target utterances, 1,985 bona fide non-target utterances and 63,882 deepfake utterances generated from seven TTS and six voice conversion (VC) algorithms. We only use the bona fide utterances in the training split to train our one-class models.

ASVspoof 2021 extends the work of ASVspoof 2019 [204]. The ASVspoof 2019 used clean speech signals. Subsequent studies demonstrated that the proposed deepfake detection systems degraded in the presence of noisy data [164]. The ASVspoof 2021 LA task introduced noisier bona fide and spoofed speech to address this discrepancy. They added noise by transmitting utterances from both classes through telephony systems. ASVspoof 2021 does not include a new training partition. Participants had to use the training split from ASVspoof 2019. We follow the same protocol in our experiments and use ASVspoof 2019's training partition for training. ASVspoof 2021's evaluation split includes 14,816 bona fide utterances and 133,360 spoofed utterances from thirteen TTS, VC and hybrid spoofing attacks. Like ASVspoof 2019, the bona fide samples were from VCTK [238]. ASVspoof 2021 also includes a new "deepfake" subset. However, we exclude it from our experiments as its evaluation database overlaps with the LA subset.

WaveFake contains 117,985 generated audio clips from six generative architectures of varying sizes [239]. The architectures are primarily generative adversarial networks (MelGAN [189], Parallel WaveGAN [240], HiFI-GAN [241] and variants), although the dataset also includes samples from WaveGlow, a flow-based generative model [242]. They use an English language dataset, LJSpeech [190], and a Japanese language dataset, JSUT [243], to create the audio. Both datasets contain a single female speaker. In total, WaveFake contains ten subsets. We use utterances from LJSpeech and JSUT as the bona fide samples. We only use the WaveFake samples for testing. We use the first 80% of the samples for training and the remainder for testing.

FMFCC is a Mandarin language dataset designed for the inaugural fake media forensic challenge of the China Society of Image and Graphics [244]. The training set contains 4,000 bona fide and 6,000 deepfake utterances. The test set contains 3,000 bona fide and 17,000 deepfake utterances. The bona fide utterances came from 58 speakers of varying ages and genders, while the deepfakes originated from thirteen TTS and VC systems.

CFAD is a Mandarin language dataset for deepfake detection research [195]. The dataset contains clean, noisy, and codec-compressed variants. We only use the clean variants for our experiments. The train split includes 12,800 bona fide utterances from four corpora

(AISHELL1, [245] AISHELL3 [246], THCHS30 [247] and MagicRead [248]) and 25,600 deepfake utterances generated from eight TTS algorithms. All deepfakes used one corpus as a basis for generation, AISHELL3 [246]. The test split divides into seen and unseen speaker subsets. We only use the seen subset to evaluate the anomaly detectors' generalisation performance on another unseen dataset. The seen split contains 14,000 bona fide and 28,000 deepfake utterances.

We disregard the deepfake samples in the training split and only use bona fide utterances for our anomaly detection experiments. We devise two training scenarios to reflect instances where the bona fide distribution is narrow and diverse:

- **Unpooled**: We only use bona fide samples from one dataset (e.g., ASVspoof 2019).
- **Pooled**: We combine the bona fide samples from all datasets (i.e., ASVspoof 2019, WaveFake, FMFCC, CFAD).

We leave out 20% of the training data for validation. We evaluate one dataset at a time (e.g., ASVspoof 2019) for both the unpooled and pooled scenarios.

We measure the generalisation capability of our models by evaluating them with the **In-The-Wild (ITW)** dataset [14]. ITW aims to reflect more realistic audio conditions. The dataset contains recordings from 58 English-speaking celebrities and politicians. In total, it has 17.2 hours of deepfake and 20.7 hours of bona fide audio. The deepfake clips came from publicly available video and audio files explicitly advertising speech deepfakes. The curators manually matched the audio data with bona fide samples from the same speaker with similar noise, emotion and duration. We do not use ITW to train any models.

6.2.2 Approach

We vary the underlying embedding by altering the architecture and fine-tuning objectives. We compare the strengths and weaknesses of different shallow detectors by training them on fixed underlying embeddings.

6.2.2.1 Architecture

We only use transformers in our experiments, as they have recently surpassed CNN performance on audio tasks [249]. We primarily use state-of-the-art representation learning architectures designed to process speech and audio. We include a vision transformer to measure how the pre-training modality affects results. Appendix B.2 contains a more detailed overview of the models.

- **HuBERT** is a self-supervised model designed for speech [216]. It is an adaptation of BERT [82]. It uses the masked prediction loss to learn the sequential structure of speech. As HuBERT used an English-language dataset for training [250], we also used a version of HuBERT[1] trained on WenetSpeech, a Mandarin-language dataset [251]. We refer to this version as HuBERT$_{\text{zh}}$ while we refer to the original version as HuBERT$_{\text{en}}$.

- **Wav2Vec 2.0** is another self-supervised speech model [218]. It uses a contrastive learning objective to learn speech representations [219]. Like HuBERT, we also vary the language. Therefore, we have two variants: Wav2vec2$_{\text{en}}$ and Wav2vec2$_{\text{zh}}$[2].

- **Audio spectrogram transformer (AST)** specialises in audio classification [249]. It takes two-dimensional spectrograms as input and splits them into overlapping patches. The authors showed that AST outperformed state-of-the-art CNNs on audio classification tasks. We use a version fine-tuned on AudioSet [252].

- **Self-supervised audio spectrogram transformer (SSAST)**: SSAST uses the same underlying architecture as AST but uses masked image modelling as the pre-training objective [253]. We use a version fine-tuned on AudioSet [252] and Librispeech [250].

- **Vision transformer (VIT)** specialises in image classification [121]. It is the inspiration behind the AST and has an identical architecture. We include VIT pre-trained on ImageNet [95] in our experiments to see if an architecture pre-trained on audio is more advantageous.

As a benchmark, we record anomaly detection performance on more traditional feature engineering approaches. We use the raw waveform, STFT, and methods designed for speech processing, including Mel spectrograms, MFCCs [254] and LFCCs [255]. We use the torchaudio library [256] with default settings to pre-process the utterances.

[1] https://huggingface.co/TencentGameMate/chinese-hubert-base
[2] https://huggingface.co/TencentGameMate/chinese-wav2vec2-base

6.2.2.2 Training objectives

Some anomaly detection works state that using embeddings from pre-trained networks is competitive [16, 84, 96]. Others state fine-tuning with the bona fide training data elicits a better, more compact embedding [67]. We investigate both approaches. Hence, we include both pre-trained and fine-tuned variants:

Pre-trained. We extract the embeddings without additional fine-tuning.

Centre loss is a common objective in one-class learning. It aims to minimise the distance between bona fide samples and a prototypical embedding [66, 67]. The loss makes the training embedding more compact. Consequently, deepfakes (which have different features) should lie further away. However, the centre loss can cause feature collapse, where all embeddings map to a single point [66, 67]. We control this by only fine-tuning for one epoch, as we found training for longer did not improve detection performance.

Open set. While fine-tuning to optimise a single class is prone to collapse, other studies show that classification models with careful hyperparameter tuning can identify OOD samples [257]. Using classification models to find OOD samples is typically called open-set detection [22]. We reconfigure the bona fide training paradigm into an open-set setup using metadata. Although gathering a sufficient variety of deepfakes for training is cumbersome, bona fide samples are easier to find. Therefore, we fine-tune the transformers to classify between the different bona fide sources depending on the metadata label.

We choose hyperparameters using random search and select based on the highest validation accuracy on the closed set classes. The hyperparameters we vary are the underlying pre-trained model (which are defined in §6.2.2.1), loss function (cross-entropy, adversarial reciprocal points learning (ARPL), and additive angular margin (AAM)), and learning rate. Some previous literature claims a specialised loss function improves OOD detection on images [258] whereas others claim cross-entropy is sufficient [257]. We investigate whether this is the case for audio data.

We fine-tune for one epoch only for comparability with the pooled centre loss arrangement. The open set training tasks are as follows:

- **Dataset source**. We train the model to classify between the bona fide data sources (*ASVspoof 2019* [163], *FMFCC* [244], *AISHELL1* [245], *AISHELL3* [246], *MagicRead*

[248], *THCHS30* [247], *LJSpeech* [190], *JSUT* [243]). There are eight classes in total.

- **Language**. We train the model the model to classify between the three source languages (*English, Mandarin, Japanese*).
- **GMM clusters**. Previous research found that pre-trained speech models can group speakers despite not being specifically trained to do so [232, 259]. We investigate whether this characteristic enables fine-tuning. We pass the training data through the pre-trained transformer and extract the embeddings. We use the embeddings to train a GMM. We set the number of Gaussian components using hyperparameter selection. We then assign the labels for each training datum using the GMM. Finally, we use the GMM labels to fine-tune the model.

6.2.3 Anomaly detection

Once we have trained the transformers, we pass the data through them and extract embeddings after each fully connected layer. We choose to include all hidden embeddings instead of solely the pre-logits layer, as other works show anomalies may manifest in different layers [46, 94]. For example, HuBERT$_{\text{base}}$ has thirteen blocks, so we obtain a stack of thirteen embeddings for one datum. We concatenate the embeddings from each layer and mean pool in the time axis. Literature suggests mean pooling benefits downstream tasks [260]. We generate embeddings for training and test data in the same way.

We use the training embeddings to construct various shallow anomaly detectors. These include k-NN [16], cosine similarity (cosine) [133], isolation forest (iForest) [261], Mahalanobis distance [96], and residual norms (norms) [262].

6.3 Results

We commence by analysing detector performance in §6.3.1. We start with detector performance to make the anomaly scoring approach consistent for the subsequent sections. We then identify the best-performing embeddings across datasets and training regimes. §6.3.2 analyses the effect of different ablations on performance. We attempt to understand why some embeddings are better than others by looking at ways to measure embedding quality in §6.3.3. We also break results down by dataset to see if there are any trends. Finally, we compare one-class anomaly detection performance to supervised classifiers in §6.3.4.

6.3.1 Overall performance

6.3.1.1 Cosine similarity is the best detector, but not by a significant margin

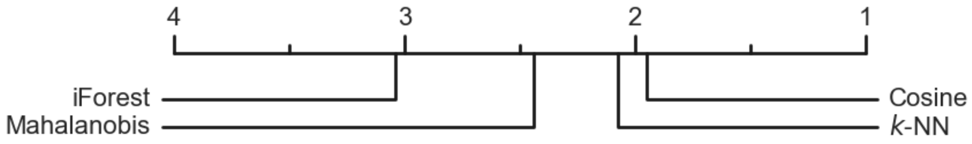


Figure 6.2: Critical difference diagram comparing the detectors in a pairwise manner. The horizontal scale denotes the average rank of each detector. Statistical differences ($p < 0.05$) would be denoted by horizontal lines between the ranks. We did not identify statistical differences between any of the detectors.

Figure 6.2 ranks the detectors, fixed by dataset, architecture and fine-tuning mode. Cosine similarity ranks the highest. But, there is no statistical difference between the methods. This ranking suggests the detectors assess test data similarly, and the choice of detector is not as important as other design choices. Nonetheless, the superior performance of cosine similarity and *k*-NN indicate that a reasonable representation clusters bona fide samples tightly and in the same direction.

We report the following results using cosine similarity as the detector. Although other speech deepfake works report equal error rate[3] or F1 scores, we found that AUROC correlates highly with these metrics[4]. Therefore, we use AUROC as the performance metric to ensure consistency across the thesis.

6.3.1.2 Deepfakes are detectable with one-class anomaly detection

We aggregate the AUROC scores for all datasets to isolate the impact of the pre-trained embeddings. Figure 6.3 shows that the pre-trained embeddings can elicit scores greater than random performance. However, AST performs the best. After AST, SSAST is the best embedding, followed by VIT, the Wav2vec2 models, and the HuBERT models. These results suggest that audio-specific features are better than general visual features. However, using embeddings trained from solely bona fide speech is insufficient for drawing out discriminative features unique to deepfakes. The deep embeddings also outperform the shallow embeddings, potentially as deep embeddings are better at capturing hierarchical features

[3] EER - the percentage at which the false positive and false positive rates are equal. Values closer to zero are more desirable.

[4] Our results showed a correlation of -0.99 between AUROC and EER and a correlation of 0.93 between AUROC and F1.

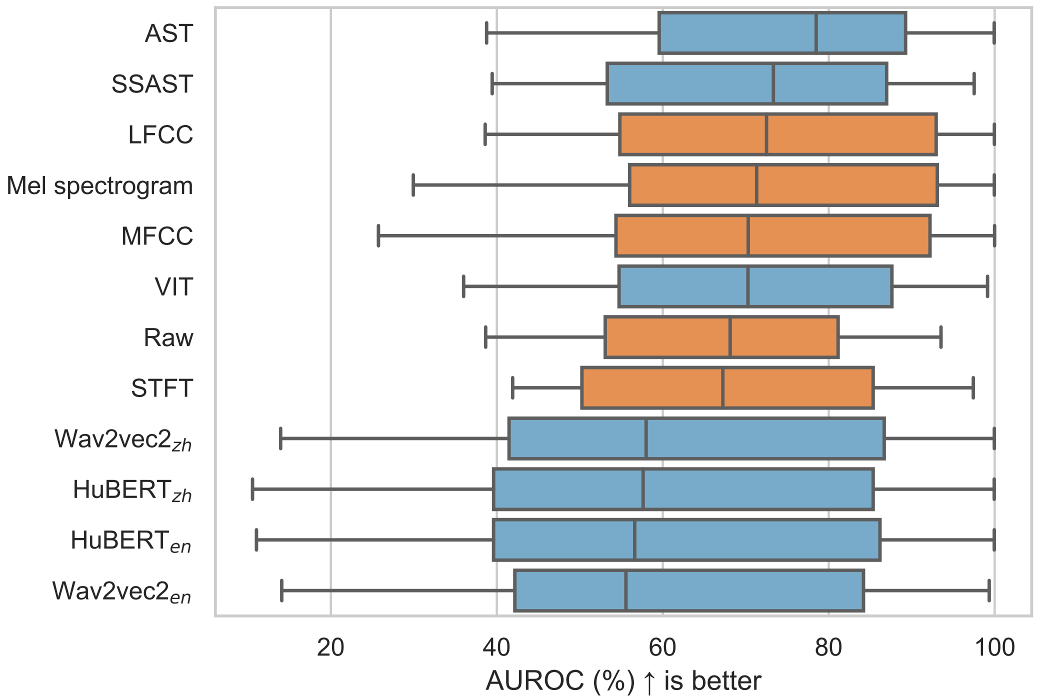


Figure 6.3: Box plot comparing cosine AUROCs for each pre-trained and baseline embedding, ordered by median performance. The orange box plots are raw embeddings and the blue box plots are neural embeddings.

lost in shallow feature engineering. These results confirm neural embeddings might be better frontend features for detection systems [215].

6.3.2 Ablation studies

We proceed to investigate how detection performance shifts under fine-tuning and hyperparameter changes.

6.3.2.1 Fine-tuning with a centre loss achieves better results, although performance depends on the diversity of the training data

Figure 6.4 shows how fine-tuning affects performance. Fine-tuning offers benefits over the pre-trained embedding because it allows the bona fides to fit a more compact space. This advantage is most prominent when training unpooled datasets with centre loss.

However, fine-tuning results highly correlate with pre-trained performance (Figure 6.5). Therefore, it is essential to choose an appropriate pre-trained model for fine-tuning. Centre loss performance declines in the pooled scenario, where the bona fide data has more diverse properties. Grouping different distributions masks these diverse characteristics. In

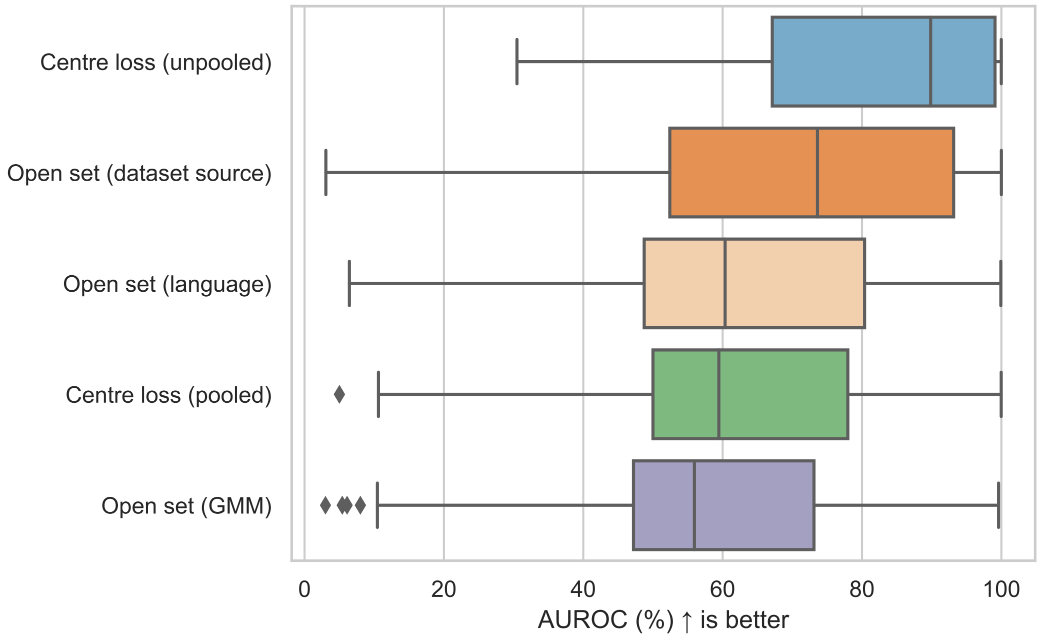
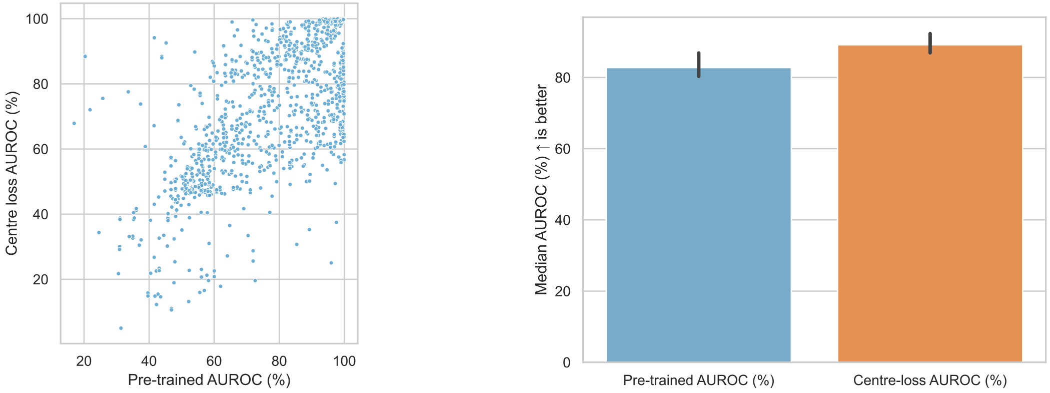


Figure 6.4: Box plot comparing cosine AUROCs for each fine-tuning method.



(a) Scatter plot comparing pre-trained and centre loss AUROCs. $\rho = 0.69$.

(b) Bar chart comparing median pre-trained and centre loss AUROCs.

Figure 6.5: Charts comparing pre-trained and centre-loss AUROCs grouped by dataset and underlying embeddings on unpooled and pooled bona fide data.

this case, training with an open set objective is more beneficial.

Predicting the dataset source is better than using language or GMM clusters for labelling. This result may be due to differing recording environments for each dataset, which is more important than the source language or content of the utterances.

Although GMM is slightly better than random, it assumes the bona fide distributions are Gaussian. Without learning to correct the clusters through another learning mechanism, such clusterings may give erroneous assignments [263].

Following suggestions that pre-processing can increase the discriminative features between bona fide and spoof utterances [213], we investigate whether this is also the case for anomaly detection.

6.3.2.2 Standardisation does not improve deepfake detection

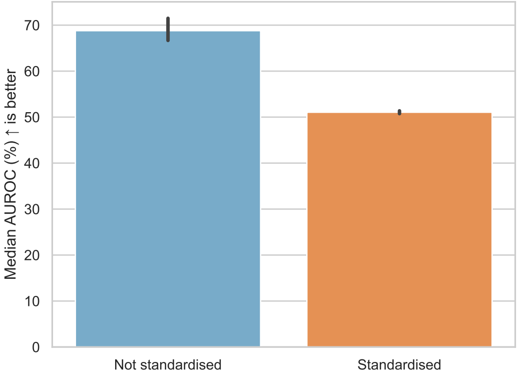


Figure 6.6: Bar chart showing how standardisation affects detection performance without RawBoost, grouped by embedding and dataset on unpooled and pooled bona fide data.

OOD detection strategies in the vision domain often require standardised embeddings [16, 134]. The assumption is that two samples from the same distribution should map to similar features [148] and the anomaly scorers should focus on the underlying data structure rather than irrelevant variations like magnitude or scale. We investigate whether this is necessary for speech deepfake detection.

We extract the embeddings per Figure 6.1 and standardise each dimension independently by removing the mean and scaling to unit variance. Figure 6.6 compares anomaly detection performance on standardised and non-standardised features. Standardising the data leads to a significant decrease in mean performance.

6.3.2.3 Data augmentation can boost deepfake detection for more specialised pre-trained embeddings

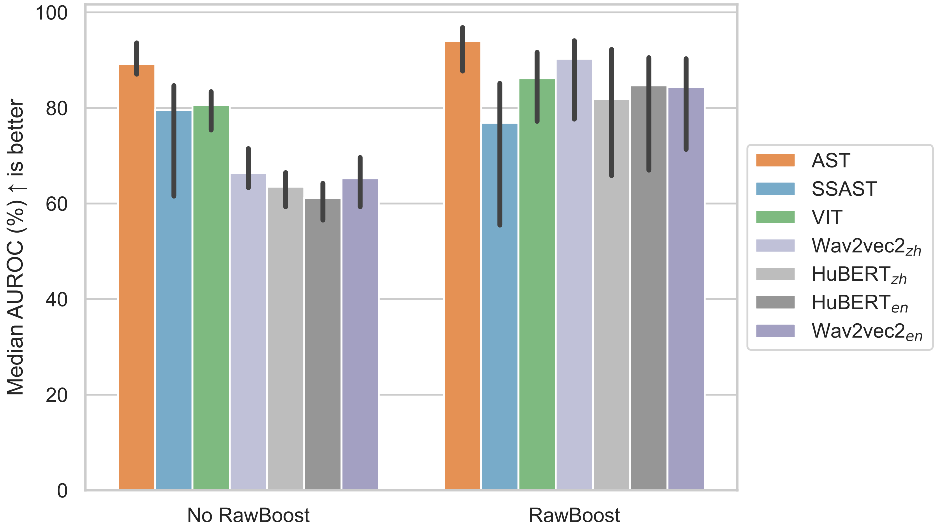


Figure 6.7: Bar chart showing how RawBoost affects detection performance without standardisation, split by underlying embedding.

Previous studies showed that spoof detectors trained on ASVspoof 2019 did not label noisy deepfakes suspicious [164], as the noise degraded the learnt discriminative features. This weakness was partly the motivation for developing ASVspoof 2021 [165]. Works suggest diversifying the training embeddings with augmentations might improve generalisation to bona fides, as the models will overfit less to irrelevant features like noise.

We investigate whether data augmentation improves generalisation to bona fides in the one-class setting and, consequently, deepfake detection. We augment the input waveforms with RawBoost [264] as it does not require external data sources. In addition, the method was designed specifically for anti-spoofing to mimic telephony scenarios. Although augmentations like pitch shift and SpecAugment can boost performance on other audio-related tasks [265], they might erode discriminative deepfake features. As an alternative, RawBoost emulates noise from encoding, transmission, microphones, and amplifiers through convolutive and additive noise.

We use the best-performing parameters as reported in the original paper. We apply convolutive noise followed by impulsive signal-dependent additive noise. We then follow the extraction protocol per Figure 6.1.

Figure 6.7 compares anomaly detection performance on augmented and non-augmented data. RawBoost has a positive effect overall. This impact is greatest for the models specialising in speech (HuBERT and Wav2Vec2), although the more general-purpose models also experience a small benefit.

6.3.3 Dataset and embedding analysis

6.3.3.1 Per dataset analysis

We seek to understand why some deepfakes are more detectable than other types and if there are differences based on the underlying synthesis method. Table 6.1 displays the ranking for all datasets for each pre-training embedding. We include full AUROC scores in the Appendix (Table C.5 and Table C.6).

Table 6.1: Rankings of each dataset (1 - highest scoring dataset, 69 - lowest scoring dataset) by pre-trained cosine AUROC for each embedding.

Dataset	AST	SSAST	VIT	Wav2vec2$_{zh}$	HuBERT$_{zh}$	HuBERT$_{en}$	Wav2vec2$_{en}$
ASVspoof 2019 A07	22	3	4	19	14	12	6
ASVspoof 2019 A08	19	14	15	17	17	16	9
ASVspoof 2019 A09	14	23	2	15	8	11	13
ASVspoof 2019 A10	31	6	7	20	15	13	7
ASVspoof 2019 A11	11	11	6	13	6	8	5
ASVspoof 2019 A12	32	19	8	22	16	15	15
ASVspoof 2019 A13	23	1	1	14	1	6	3
ASVspoof 2019 A14	27	22	16	18	18	14	8
ASVspoof 2019 A15	38	26	21	23	22	18	20
ASVspoof 2019 A16	44	17	19	25	21	19	17
ASVspoof 2019 A17	49	48	57	53	49	45	38
ASVspoof 2019 A18	16	51	45	28	27	29	27
ASVspoof 2019 A19	54	46	52	54	55	56	55
ASVspoof 2021 A07	37	27	44	35	34	31	31
ASVspoof 2021 A08	42	34	50	36	37	33	33
ASVspoof 2021 A09	33	38	42	33	31	30	37
ASVspoof 2021 A10	47	29	41	38	35	32	32
ASVspoof 2021 A11	28	31	40	26	30	28	29

Continued on next page

Table 6.1: Rankings of each dataset (1 - highest scoring dataset, 69 - lowest scoring dataset) by pre-trained cosine AUROC for each embedding. (Continued)

ASVspoof 2021 A12	53	35	46	41	39	34	36
ASVspoof 2021 A13	41	25	32	29	28	26	28
ASVspoof 2021 A14	56	41	49	37	41	36	40
ASVspoof 2021 A15	63	42	51	42	46	39	43
ASVspoof 2021 A16	60	36	48	46	44	40	35
ASVspoof 2021 A17	59	57	60	59	58	53	48
ASVspoof 2021 A18	35	58	53	50	50	47	46
ASVspoof 2021 A19	62	53	61	60	63	62	56
CFAD AISHELL1 F01	25	20	20	31	29	24	41
CFAD AISHELL1 F02	15	5	17	27	32	38	44
CFAD AISHELL1 F03	30	37	18	47	45	43	49
CFAD AISHELL1 F04	21	16	26	24	25	27	34
CFAD AISHELL1 F05	13	43	25	12	23	20	14
CFAD AISHELL1 F06	20	7	23	34	38	35	50
CFAD AISHELL1 F07	17	55	24	21	24	21	19
CFAD AISHELL1 F08	9	39	14	9	20	17	11
CFAD AISHELL3 F01	43	45	55	51	52	55	58
CFAD AISHELL3 F02	24	13	30	32	42	51	54
CFAD AISHELL3 F03	36	69	31	48	43	49	57
CFAD AISHELL3 F04	26	33	35	30	26	42	42
CFAD AISHELL3 F05	12	52	27	7	11	23	18
CFAD AISHELL3 F06	39	21	34	49	54	54	59
CFAD AISHELL3 F07	18	65	28	16	19	25	21
CFAD AISHELL3 F08	10	49	22	6	10	22	16
CFAD MagicRead F01	68	32	59	57	65	59	53
CFAD MagicRead F02	50	4	36	45	51	48	47
CFAD MagicRead F03	69	56	56	58	56	57	52
CFAD MagicRead F04	58	15	43	44	47	46	39
CFAD MagicRead F05	48	40	39	39	36	41	25

Continued on next page

Table 6.1: Rankings of each dataset (1 - highest scoring dataset, 69 - lowest scoring dataset) by pre-trained cosine AUROC for each embedding. (Continued)

CFAD MagicRead F06	66	9	37	55	57	52	51
CFAD MagicRead F07	55	47	38	43	40	44	30
CFAD MagicRead F08	34	44	33	40	33	37	23
CFAD THCHS30 F01	3	28	10	8	13	7	22
CFAD THCHS30 F02	1	2	3	1	4	1	10
CFAD THCHS30 F03	8	30	29	11	12	10	26
CFAD THCHS30 F04	4	24	11	4	7	4	12
CFAD THCHS30 F05	6	12	13	3	3	3	1
CFAD THCHS30 F06	5	18	5	10	9	9	24
CFAD THCHS30 F07	7	10	12	5	5	5	4
CFAD THCHS30 F08	2	8	9	2	2	2	2
FMFCC	67	59	58	69	69	69	69
WaveFake JSUT Multi Band MelGAN	64	66	63	66	66	63	62
WaveFake JSUT Parallel WaveGAN	65	54	67	62	62	64	68
WaveFake LJSpeech Conformer	29	63	54	52	48	50	45
WaveFake LJSpeech Full Band MelGAN	52	64	65	67	67	66	67
WaveFake LJSpeech HiFi-GAN	57	62	69	68	68	68	66
WaveFake LJSpeech MelGAN	61	68	62	61	61	67	65
WaveFake LJSpeech MelGAN Large	40	67	47	64	60	65	64
WaveFake LJSpeech Multi Band MelGAN	46	60	64	63	59	58	61
WaveFake LJSpeech Parallel WaveGAN	51	50	68	65	64	61	63

Continued on next page

Table 6.1: Rankings of each dataset (1 - highest scoring dataset, 69 - lowest scoring dataset) by pre-trained cosine AUROC for each embedding. (Continued)

WaveFake Waveglow	LJSpeech	45	61	66	56	53	60	60

The rankings of datasets suggest that language is irrelevant to detectability, indicating the synthesis method and the quality of the original bona fide data is the cause for the varying difficulty. The language used for model pre-training does not affect the detection performance of datasets in different languages. We can take CFAD THCHS30 F02 [195, 247] as an example. This type of deepfake was generated using a Griffin-Lim [266] vocoder from a Mandarin language dataset comprising forty speakers recorded using carbon microphones in an office. This dataset was the easiest to detect for most embeddings, including those trained using Mandarin and those trained using English data.

There were similarities between the easiest and hardest-to-detect datasets across the neural architectures and fine-tuning tasks, suggesting deep architectures capture similar discriminative features and make similar labelling mistakes. For instance, ASVSpoof2019 13 (where spoofs come from a combined neural network-based voice conversion and TTS system) obtained an average AUROC of 98.85% across pre-trained embeddings, 99.67% on the centre-loss models, 98.82% on the metadata open set models, 90.25% on the language open set models and 84.67% on the GMM models.

In contrast, all struggled to detect CFAD MagicRead F03, a neural vocoder-based system designed for low-power devices. It achieved an mean AUROC of 57.72% across the pre-trained embeddings, 60.19% on the centre-loss models, 27.05% on the metadata open set models, 14.96% on the language open set models, and 32.30% on the GMM models. Overall, CFAD THCHS subsets were the easiest to detect, while WaveFake and FMFCC were the most challenging.

Although there are similarities across the neural architectures, the difficulty ranking across the shallow embeddings differs. For instance, the LFCC embedding achieved a mean AUROC of 90.85% on CFAD AISHELL3 F06[5], whereas the centre loss embeddings averaged 83.89%.

[5] The bona fide dataset was AISHELL3 and the deepfakes were made using HiFiGAN (a neural vocoder)[241], using AISHELL3.

We also visualise how similar the datasets are in the neural embeddings using dendrograms. We follow the methodology in the ASVspoof 2019 paper to make the dendrograms [182].

We extract the test embeddings per Figure 6.1. We use the test embeddings as we include deepfakes in our analysis. We whiten embeddings using the mean and standardisation of each dataset and unit-normalise them. We then compute the pairwise cosine similarities between attacks using the following equation:

$$D(\boldsymbol{X}_i, \boldsymbol{X}_j) = \frac{1}{|\boldsymbol{X}_i|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_i} \min_{\boldsymbol{y} \in \boldsymbol{X}_j} \big(1 - s(\boldsymbol{x}, \boldsymbol{y})\big) \tag{6.1}$$

where $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ denote the collections of vectors in distinct datasets, $s(\boldsymbol{x}, \boldsymbol{y})$ is the cosine similarity between embeddings $\boldsymbol{x}$ and $\boldsymbol{y}$, and $|\boldsymbol{X_i}|$ denotes the number of vectors in $\boldsymbol{X_i}$. We split out the bona fide and deepfake subsets, resulting in 53 datasets (8 bona fide and 45 deepfake). For CFAD, all deepfakes originate from AISHELL3. and the spoof datasets are reused for the different CFAD subsets. Consequently, we get a 53×53 distance matrix. We run agglomerative clustering on the matrix to get the final visualisations.

Figure 6.8 shows the results of the AST embeddings. We noticed similar trends across the architectures. TTS deepfakes (e.g. ASVspoof 2019 and 2021 A08 to A14) tend to be distant from their bona fide counterparts, while VC deepfakes (ASVspoof 2019 A17 to A19) and more traditional spoofs were closer. For CFAD, the traditional deepfakes (F01 and F02) were closer to the AISHELL3 bona fide set, while the GAN-based methods (F05, F07, F08) were further away. We note WaveFake uses a subset of the same neural synthesis methods as CFAD, such as MelGAN and parallel WaveGAN. However, WaveFake's deepfakes lie closer to the bona fide samples. This behaviour suggests the nature of the bona fide data also affects detection difficulty.

Based on this analysis, the following factors appear to influence the difficulty of detecting speech deepfakes:

1. **Speech synthesis method**: TTS methods are more distinct from bona fide than VC methods.

2. **Noise**: THCHS30 bona fide speakers were the easiest to identify across the models.

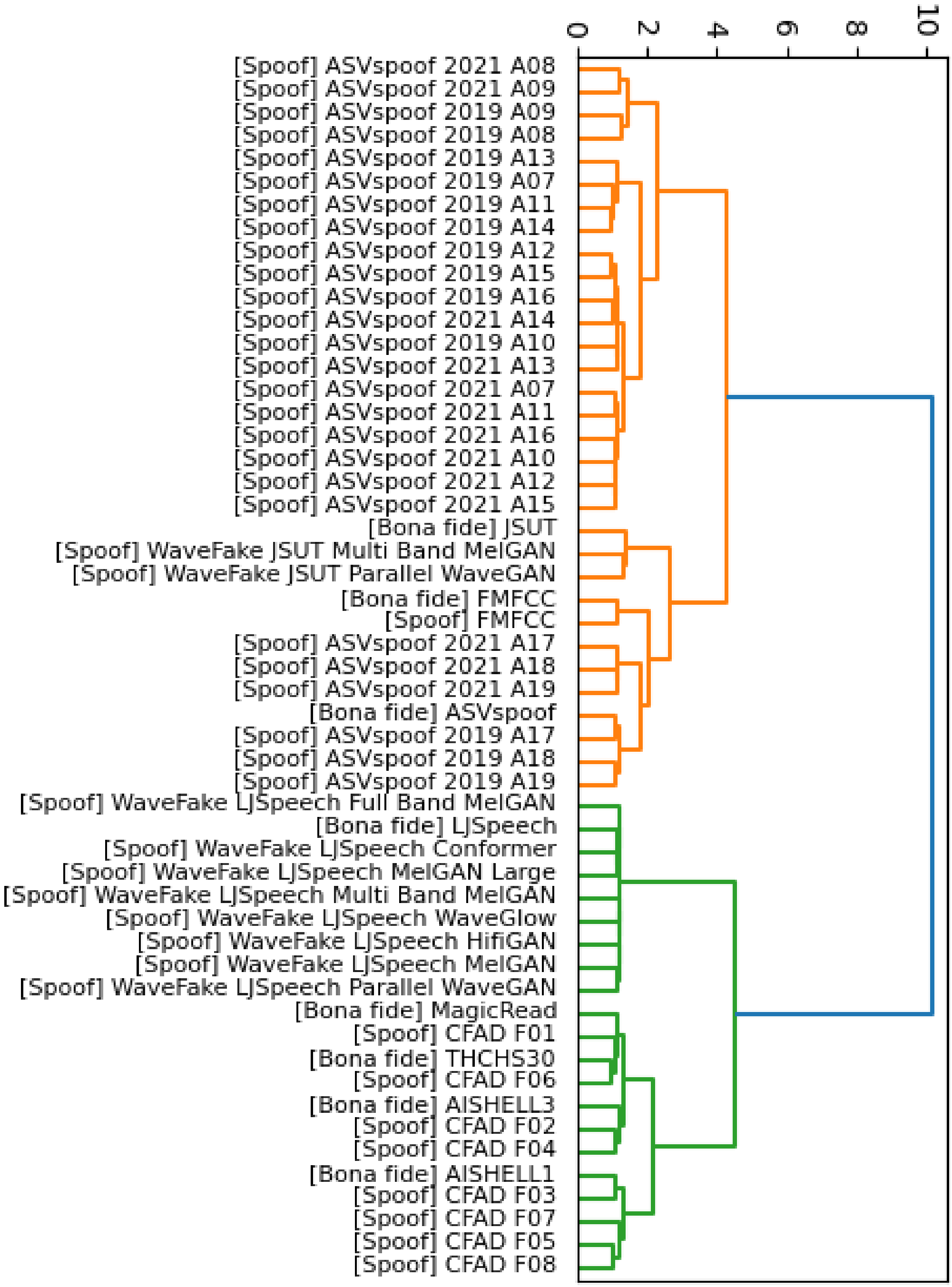
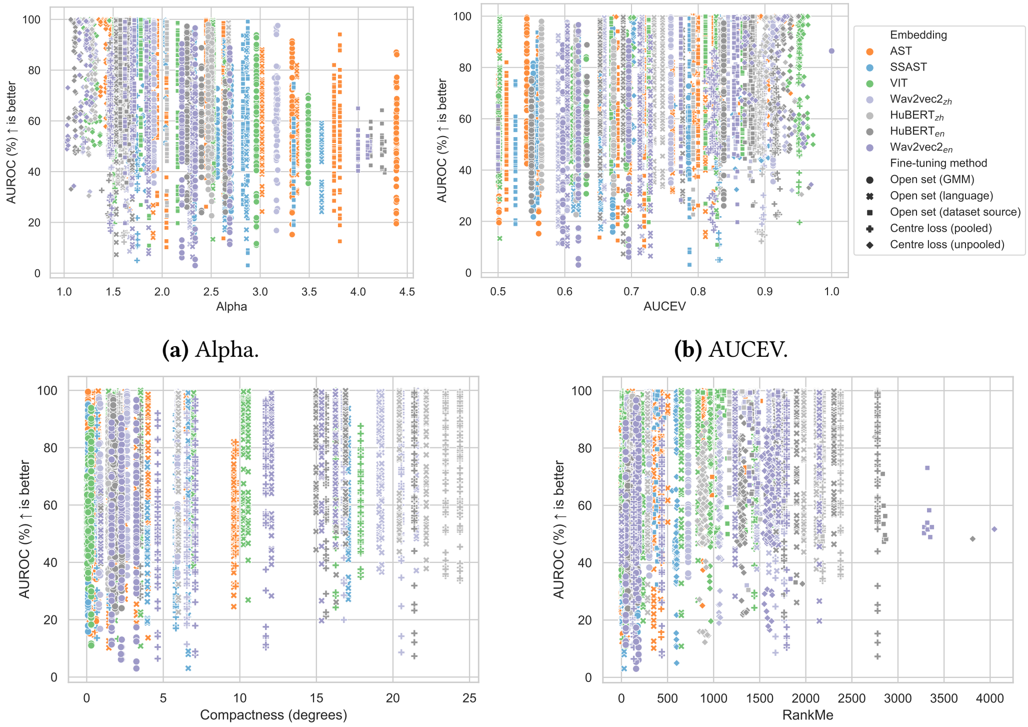


Figure 6.8: Agglomerative clustering of the bona fide and deepfake subsets using pre-trained AST embeddings.

As a carbon microphone recorded the speakers, the bona fide recordings might contain artefacts distinct from the deepfakes. However, an attacker could circumvent this issue by adding noise and compression to the deepfakes.

3. **The number of bona fide speakers to attack**: WaveFake only contained samples from one speaker, unlike the other datasets. A smaller number of speakers would make it easier for a synthesis method to mimic the targets' characteristics.

6.3.3.2 No reliable method to measure embedding quality exists



(a) Alpha.

(b) AUCEV.

(c) Compactness.

(d) RankMe.

Figure 6.9: Scatter plots comparing measures of embedding quality against AUROCs for all datasets, grouped by underlying pre-trained embedding and fine-tuning regime.

As our approach involves multiple hyperparameters, we investigate whether there is a way to quantify embedding quality to aid hyperparameter selection. Measuring embedding quality is important for anomaly detection as our validation sets only contain benign data. We implement several measures that claim to link downstream performance with unsupervised learning. These measures are (described in more detail in §2.2.2):

- **Alpha** [59]: A covariance-based method that claims lower values signal more sensitivity to small changes in input stimuli. Values closer to 1 are desirable.

- **Area under cumulative explained variance (AUCEV)** [61]: The cumulative explained variance of the singular values. AUCEV measures the extent of dimension collapse. An AUCEV of 1 indicates severe dimensional collapse, whereas an AUCEV of 0.5 suggests no collapse. AUCEVs closer to 0.5 are better.

- **Compactness** [68]: Average cosine similarity between each bona fide embedding and the mean bona fide prototype. Lower values are preferred.

- **RankMe** [60]: Estimated embedding rank. Larger values are better because they exhibit less dimensional collapse.

Figures 6.9a to 6.9d compare anomaly detection AUROC against the above four measures. We group results from all datasets, architectures, and fine-tuning modes.

There is no correlation between anomaly detection performance and any of the measures. Values correspond more closely to the underlying pre-trained model and the fine-tuning mode.

There is also no relationship between anomaly detection performance and AUROC when filtering by model and fine-tuning method. For instance, when looking at centre loss AST only, the correlations are -0.07, -0.04, 0.15 and 0.01 between anomaly detection and alpha, AUCEV, compactness, and RankMe respectively.

6.3.3.3 Discriminative features lie in low-dimensional subspaces

The success of rudimentary anomaly detectors indicates discriminative features lie in a low-dimensional subspace. We investigate this by ablating the residual norm approach [262]. A.1.1 explains the details of this method.

We look at how anomaly detection performance changes when varying the proportion of features ablated because the absolute dimensionalities depend on the architecture. The percentages we examined were {0.2%, 0.5%, 1%, 2%, 4%, 8%, 16%, 33%, 67%, 99%}.

Figure 6.10 summarises the results for all datasets on the unpooled centre loss models. Surprisingly, the residual norm approach is on par with the shallow anomaly detectors even at the smallest dimensionalities. Across datasets and models, the median cosine AUROC was 89.85%, whereas the median AUROC when using 0.02% of residuals was 84.80%. This proportion of residuals exhibits the highest AUROC scores, reinforcing the idea that the

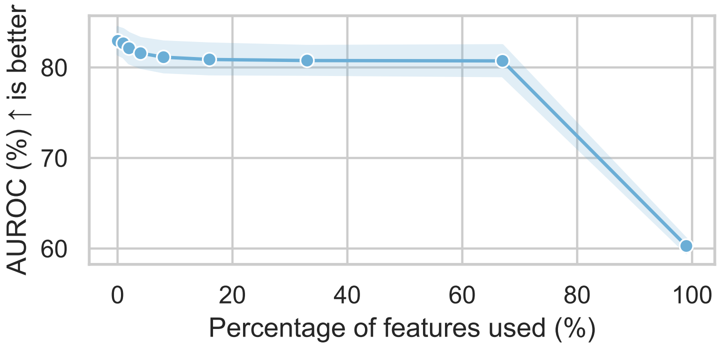


Figure 6.10: Line plot comparing residual norm AUROC scores against the percentage of features using the residual norm method.

discriminative features are low-dimensional. The plateau at higher dimensionalities also suggests this.

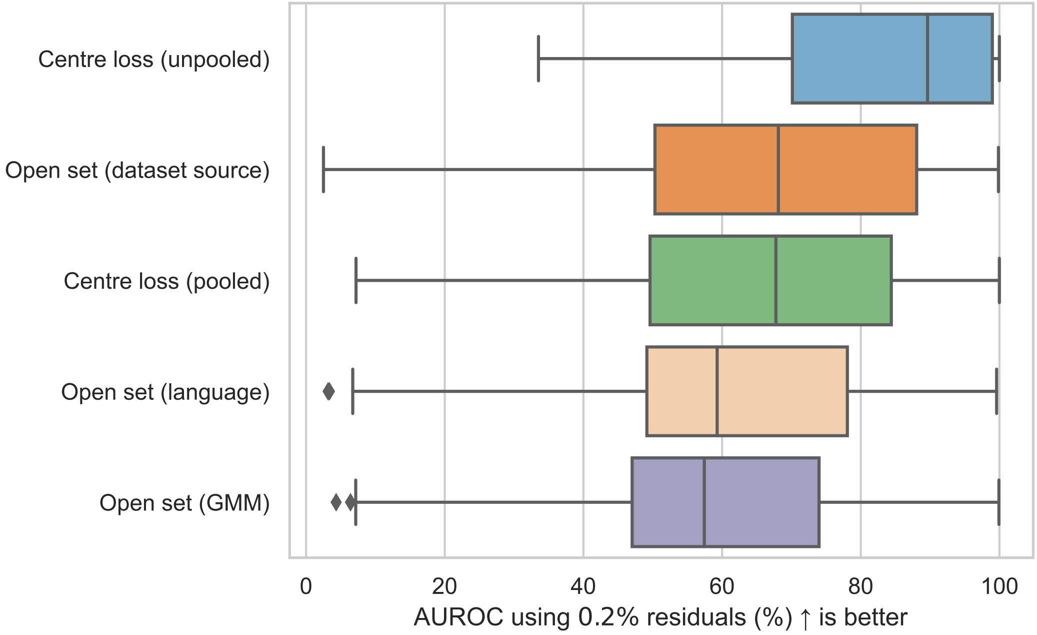
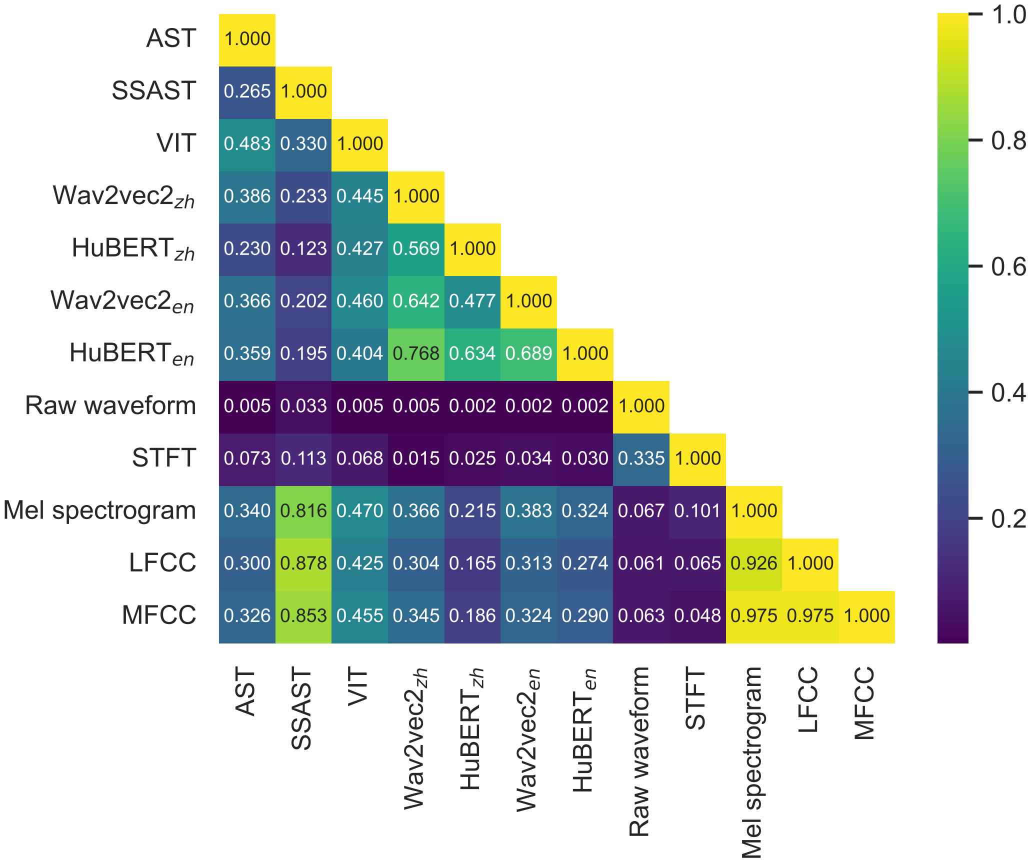


Figure 6.11: Box plot comparing residual norm AUROC scores using 0.02% of residuals per fine-tuning objective.

The centre loss objective encourages low-dimensional embeddings. Figure 6.11 compares AUROC scores using 0.02% of residuals for the different training objectives. Open set performance is worse. The shallow feature-engineered embeddings do not improve using the residual norm approach either.



	AST	SSAST	VIT	Wav2vec2$_{zh}$	HuBERT$_{zh}$	Wav2vec2$_{en}$	HuBERT$_{en}$	Raw waveform	STFT	Mel spectrogram	LFCC	MFCC
AST	1.000											
SSAST	0.265	1.000										
VIT	0.483	0.330	1.000									
Wav2vec2$_{zh}$	0.386	0.233	0.445	1.000								
HuBERT$_{zh}$	0.230	0.123	0.427	0.569	1.000							
Wav2vec2$_{en}$	0.366	0.202	0.460	0.642	0.477	1.000						
HuBERT$_{en}$	0.359	0.195	0.404	0.768	0.634	0.689	1.000					
Raw waveform	0.005	0.033	0.005	0.005	0.002	0.002	0.002	1.000				
STFT	0.073	0.113	0.068	0.015	0.025	0.034	0.030	0.335	1.000			
Mel spectrogram	0.340	0.816	0.470	0.366	0.215	0.383	0.324	0.067	0.101	1.000		
LFCC	0.300	0.878	0.425	0.304	0.165	0.313	0.274	0.061	0.065	0.926	1.000	
MFCC	0.326	0.853	0.455	0.345	0.186	0.324	0.290	0.063	0.048	0.975	0.975	1.000

Figure 6.12: Heatmap comparing dCKA values between embeddings.

6.3.3.4 Deep embeddings learn essential speech recognition features

Figure 6.12 compares dCKA values to measure the similarities between learnt embeddings. Values closer to 1 suggest higher similarities. We use an aggregate dCKA score for comparability with the shallow feature-engineering embeddings. AST exhibits moderate similarities with the different embeddings, except for raw waveforms. dCKA values between raw waveforms and the other pre-trained embeddings are also low. These results suggest combining shallow features with pre-trained embeddings could counteract the shortcomings of the neural architectures.

When comparing dCKA values between pre-trained architectures and the other shallow embeddings, there are higher similarities with the features designed for speech recognition (Mel spectrograms, LFCCs and MFCCs) compared to STFT and the raw waveforms. This observation suggests that the neural architectures learn features more specific to core speech recognition.

6.3.4 Generalisation to unseen data

6.3.4.1 General-purpose features are better for generalisation to unseen data

The previous experiments concentrate on anomaly detection using bona fide samples similar to those seen at training. We investigate if our anomaly detection approach can generalise to unseen bona fide and deepfake utterances. Initial experiments with ITW showed that high-performing classifiers trained to detect ASVspoof deepfakes degrade from 200% to 1,000% [14]. We analyse the robustness of our anomaly detectors on ITW.

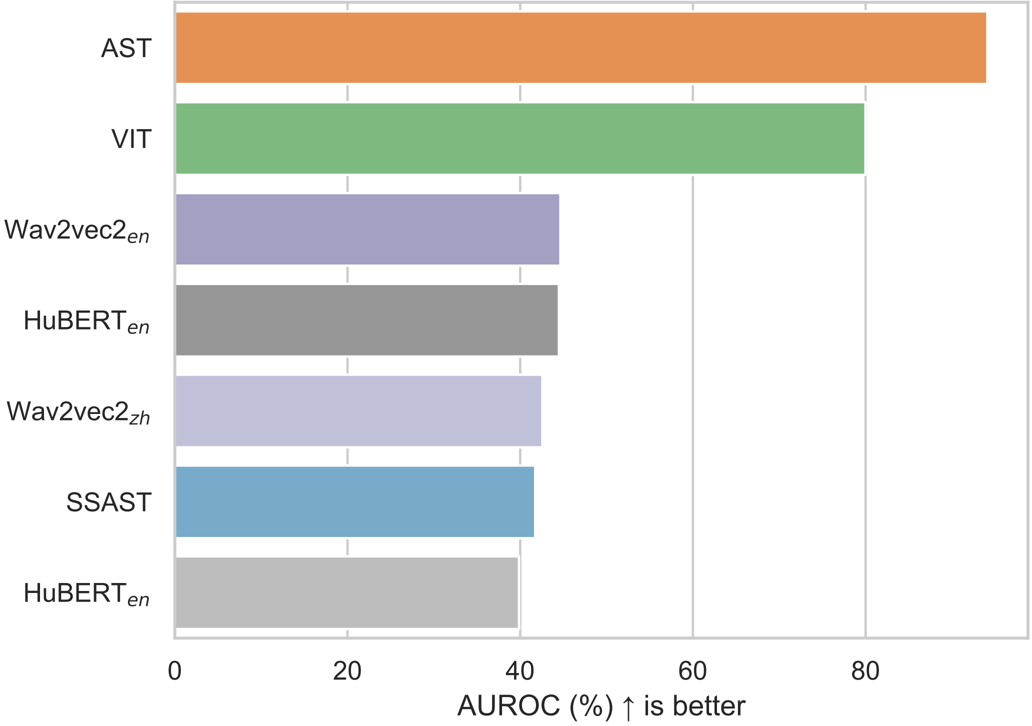
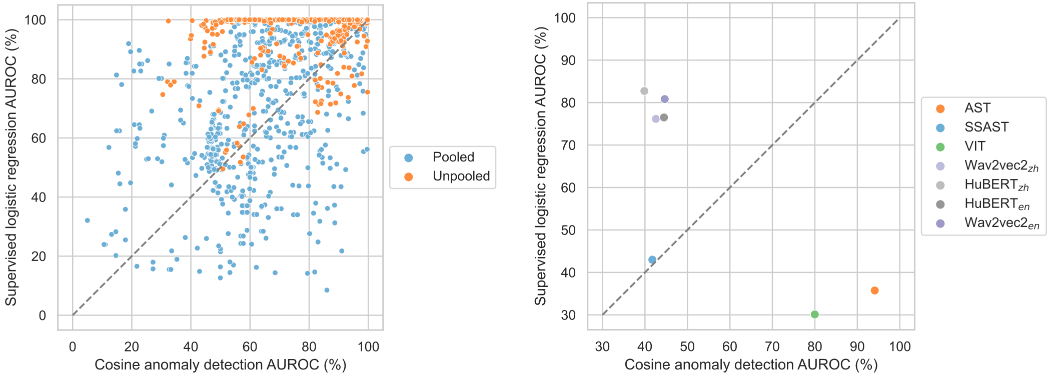


Figure 6.13: Bar chart comparing the anomaly detection performance of different pooled centre-loss fine-tuned embeddings on ITW, a dataset containing unseen bona fide and deepfake utterances.

Figure 6.13 is a box plot of cosine AUROC scores on centre-loss fine-tuned embeddings. It shows that most embeddings experience a performance degradation on ITW. However, AST and VIT can detect speech deepfakes with relative reliability, with AUROC scores of 94.10% and 79.99% respectively (or EERs of 11.15% and 27.84%). This performance is much higher than the supervised models reported by Müller et al. [14], where the best performing supervised embedding (RawNet 2 [213]) achieved an EER of 33.94%. Our results indicate that one-class anomaly detectors may be more stable under domain shifts. Our results also reinforce that speech-specific models may overfit to in-distribution features, whereas more general-purpose features are more robust.

6.3.4.2 Supervised classifiers outperform one-class detectors, except under bona fide distribution shifts



(a) On seen bona fide data.

(b) On ITW data.

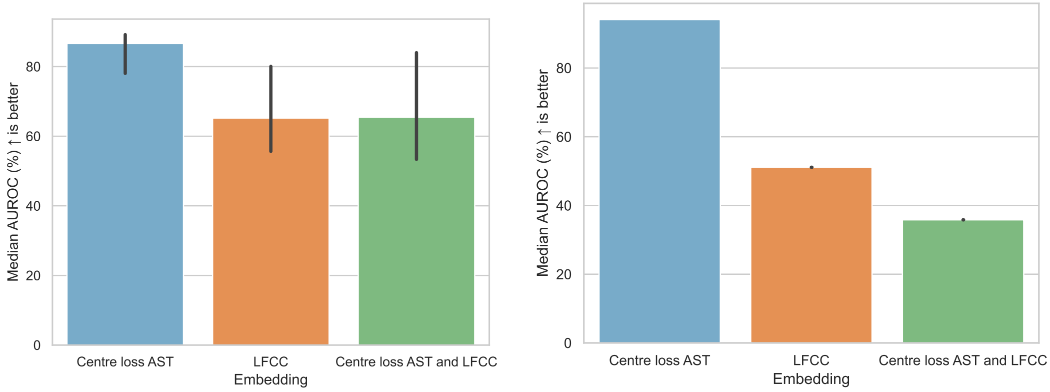
Figure 6.14: Scatter plots comparing cosine anomaly detection AUROCs against supervised logistic classifiers on centre-loss fine-tuned embeddings. The dotted $x = y$ line denotes where the anomaly detectors and supervised classifiers perform at parity.

Although deepfake detection using one-class classifiers is possible, we want to understand how they compare to supervised equivalents. We do not use an explicit backend in our supervised setup to ensure we only study the embedding and not the ability of the supervised classifiers. In particular, we extract the embeddings in the same way as the one-class protocol. We use these embeddings to train linear classifiers and random forests. We train the supervised classifiers using the provided annotated bona fide and deepfake samples in the training set, so there is no dataset leakage. When evaluating ITW, we do not include any ITW samples in the training set and instead use the in-distribution bona fide datasets (i.e., the training splits for ASVspoof 2019, WaveFake, FMFCC, CFAD). Figure 6.14 depicts the differences between the one-class and supervised setups. We report linear classification instead of random forest performance because the classification scores between the two classification methods are highly correlated ($\rho = 0.91$).

Figure 6.14a compares supervised scores to one-class detection for each architecture for the seen datasets. Supervised classifiers are an upper limit for one-class performance as they see more information (deepfake examples) at training. In most cases, supervision beats one-class learning. This difference is more prominent in the unpooled scenario.

However, we confirm anomaly detection outperforms supervised classifiers when evaluating ITW using VIT and AST (Figure 6.14b). Our results indicate that one-class strategies may be more reliable when there are bona fide distribution shifts.

6.3.5 Ensembling



(a) Excluding ITW.

(b) ITW.

Figure 6.15: Bar charts comparing cosine AUROC scores using different embeddings on pooled data.

Our dataset analysis (§6.3.3.1) and dCKA study (§6.3.3.4) suggest that neural architectures and shallow feature-engineering methods use different discriminative features. We investigate whether ensembling can use these differences to improve detection performance.

We concatenate embeddings from centre-loss AST and LFCC. We chose these embeddings because AST is the best-performing neural architecture, LFCC is relatively distinct from AST according to dCKA scores, and LFCC is the best-performing shallow feature.

Figure 6.15 compares the ensemble's results against the two embeddings individually. However, the ensemble performs worse than the individual embeddings, suggesting successful ensembling requires more sophisticated feature aggregation methods.

6.4 Conclusion

We conclude by outlining the limitations, directions for future work, and main contributions.

6.4.1 Limitations and future work

We use modality-neutral methods (centre and open set losses combined with mean-pooled embeddings) for our anomaly detection approach. The pre-trained networks vary in size and pre-training objective. Considering these variations in the anomaly detection pipeline would make comparing the pre-trained models more challenging. However, our approach does not fully exploit the expressivity of audio neural networks. Unlike modalities like images, time and frequency domains in audio could manifest discriminative features in different ways. Future work could examine the impact of time and frequency in more detail. Alternatively, studies could focus on one model to adapt the original pre-training method to deepfake detection.

In addition, we extract embeddings from all hidden layers of the models. Although previous works show no guaranteed method for identifying which intermediate embeddings work best [81], using all layers is computationally expensive. Follow-up experiments could repeat the experiments per layer to understand the networks better.

Moreover, our results suggest that shallow embeddings learn different features from the neural embeddings. Although our initial ensembling experiment was not fruitful, ensembling could rectify some weaknesses of deep pre-trained models. Additional research on the discriminative features in different speech synthesis methods and further work on fusion approaches could make one-class deepfake detection more promising.

6.4.2 Summary

This chapter studies whether one-class detectors can identify speech deepfakes. We analyse the performance of various pre-trained embeddings and the effect of fine-tuning with bona fide data. We show that one-class deepfake detection is possible, and the choice of representation space is more important than the detector. Although fine-tuning with bona fide data can help, the underlying pre-trained architecture is more indicative of downstream performance. Architectures pre-trained on more diverse datasets like audio or images are more versatile than those specially trained for speech. However, careful augmentation methods like RawBoost [264] can improve detection.

We show existing unsupervised methods that measure embedding quality are ineffective for our anomaly detection setup. However, we demonstrate that core discriminative features occupy a low-dimensional subspace in the embedding space.

Our dataset analysis indicates neural embeddings are better at identifying TTS deepfakes but struggle with VC methods. This behaviour appears to be a shortcoming of neural embeddings.

Finally, we demonstrate that one-class detectors are more robust than supervised classifiers when there is a distribution shift on bona fide data. Although supervised classifiers surpass one-class detectors on in-distribution bona fide data, one-class detectors can generalise better on ITW.

Our study highlights the value of one-class approaches for speech deepfake detection. Although no infallible solution exists, combining one-class detectors with supervised classifiers is more likely to generalise to unseen attacks.

7 | Tabular data

While self-supervised learning has improved anomaly detection in domains like images and text [113, 210], these techniques have not yielded the same benefits for tabular data [15].

Self-supervised learning uses pretext tasks to learn the intrinsic characteristics of training data in place of using labels. Images and text have spatial or sequential biases which serve as natural starting points for pretext tasks. Therefore, understanding the typical characteristics of a domain allows one to choose a suitable pretext task. One example is colourising greyscale images [30]. Colourisation requires knowledge of object boundaries and semantics. These aspects are helpful for image classification [33, 34]. In the case of text, predicting the next word in a sentence is a common choice [31, 32]. These predictions allow models to learn about grammatical structures and vocabulary. In contrast, the starting points for tabular data are unclear.

A recent study indicated that self-supervised learning does not help tabular anomaly detection [15]. Reiss et al. compared two self-supervised methods with k-NN on the original features. Even though the self-supervised 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. Firstly, we extend the experiments to include a more comprehensive suite of pretext tasks. In addition, we 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 of purely localised anomalies and those with different dependency structures.

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.

§7.1 covers the experimental approach. We evaluate our findings in §7.2. Finally, we summarise our work and conclude in §7.3. The contents of this chapter were published in Patterns Analysis and Applications [267].

Self-supervised learning and anomaly detection for tabular data

The literature covering self-supervision for anomaly detection in tabular data is more limited than in other domains like images and text. GOAD [23] extends the GEOM approach of Golan and El-Yaniv [80] from the image domain to a more generalised setting. GEOM showed that compared to OCSVMs trained on pixel space, outputs from CNNs trained to predict image rotations were more reliable for anomaly detection. GOAD applies random affine transformations to the data and trains 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 benign data versus the anomalies.

ICL [268] adapts the InfoNCE objective [39, 40]. 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 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 benign 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. [15] 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 the original implementation and their feature extraction pipelines 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.

7.1 Method

We introduce the datasets used in the study, the baseline, our main approach, and the methodology for additional ablation studies.

7.1.1 Datasets

We use 26 multi-dimensional point datasets from Outlier Detection Datasets (ODDS) [269]. Each datum comprises one record, which contains multiple attributes. Table 7.1 summarises the properties of the datasets. We treat each dataset as distinct. We train and test separate anomaly detection models for each dataset.

We follow the data split protocols described in previous tabular anomaly detection literature [23, 268]. We randomly select 50% of the benign 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 adopted a one-class setup. We partition the training set further by leaving

20% for validation.

Table 7.1: 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

7.1.2 Baseline approach

We run k-NN, iForest, LOF, OCSVM, and residual norms on the raw training data. We provide further detail on the detectors in Appendix A. Even though Reiss et al. [15] 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 §7.2.7. Another anomaly detection study, ADBench [270], follows a similar protocol but assumes anomalies are present in the training data. Our experiments establish whether a purely one-class setup affects overall detector ranking. We use scikit-learn [271] to implement all detectors except for k-NN, which uses the Faiss library [272].

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 $\{10\%, 20\%, ..., 90\%\}$. We record our findings in §7.2.7. 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 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 gave similar results.

7.1.3 Self-supervision details

This section outlines the pretext tasks, architectures, and feature extraction workflow used in the experiments.

7.1.3.1 Pretext tasks

Although tabular data lacks 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 [268] and GOAD [23] to use them as pretext tasks. We do not directly implement ICL and GOAD as they score anomalies in an end-to-end manner. 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).

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 [35, 113] 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 [273]. 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 $r \quad (r < d)$, 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. [274]. 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.

7.1.3.2 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. [275], we use ResNets [276] 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 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 [257, 258], 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) [258], and additive angular margin (AAM) [277]. 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. They define reciprocal points as "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 [68]. 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 [257]. We experiment with MSE and mean absolute error (MAE) for the autoencoders. We use the binary cross entropy (BCE) 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 7.2.

Table 7.2: Summary of the model configurations.

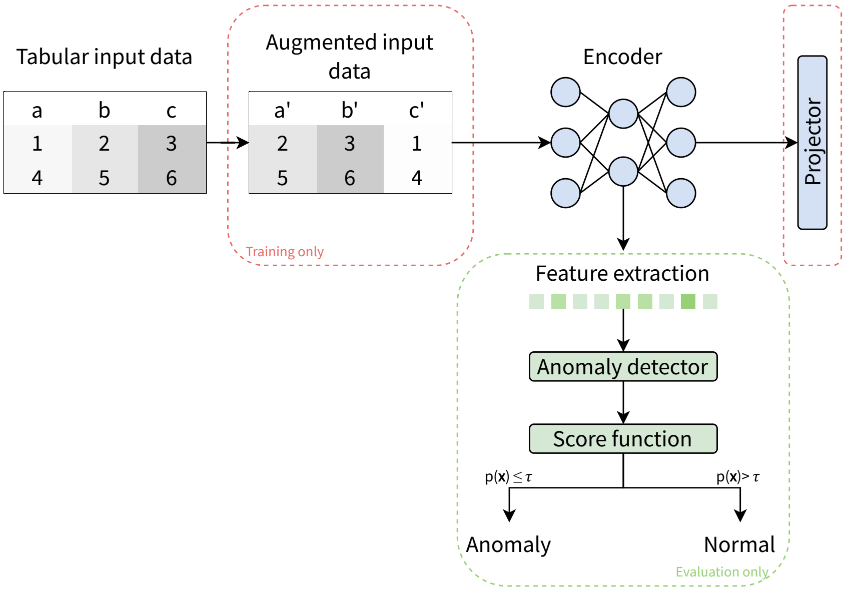
Anomaly detectors	**Architectures**	**Self-supervised tasks**	**Loss functions**
k-NN iForest LOF OCSVM Residual norms	ResNet FT-Transformer	Rotation Shuffle Mask classification	Cross-entropy ARPL AAM
		Mask columns	BCE
		Autoencoder	MSE MAE
		EICL Contrastive - rotation Contrastive - shuffle Contrastive - mask	InfoNCE VICReg

7.1.3.3 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 benign 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.

7.1.3.4 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 features. We do not apply any augmentations during inference to ensure a fair comparison between the self-supervised tasks. Figure 7.1 shows the workflow.



a	b	c
1	2	3
4	5	6

a'	b'	c'
2	3	1
5	6	4

Figure 7.1: Self-supervised anomaly detection workflow. The data are augmented and fed through the projector only during training.

7.1.4 Additional ablations

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

7.1.4.1 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 benign data in the train and test splits and only generate anomalies by perturbing the properties of the benign training data. We use the four synthetic anomaly categories as defined in ADBench [270, 278]. We use the code from ADBench to create all types.

- **Local** anomalies deviate from their local cluster. We use Gaussian mixture models (GMM) to learn the underlying benign distribution. The covariance matrix from the GMM $\hat{\Sigma}$ undergoes scaling by a factor α to generate the anomalies ($\hat{\Sigma} = \alpha\hat{\Sigma}$). We use $\alpha = 2$ in our experiments.

- **Cluster** anomalies use GMMs to learn the benign 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 benign data. We use vine copulas to learn the benign distribution and Gaussian kernel density estimators to generate anomalies. Vine copulas are graphical models that build a d-dimensional dependence structure from two-dimensional building blocks, called pair-copulas. The underlying graph structure consists of a nested sequence of trees, called vines [279].

7.1.4.2 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 [273]. We investigate if this is the case 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 is noisy or incomplete. For our ablations, we follow Grinsztajn et al. [273] and apply the following corruptions to the raw data:

1. **Adding uninformative features**: We add extra attributes to the input data **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 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 benign 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 retain 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.

7.2 Results

We organise our results as follows: §7.2.1 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. §7.2.2 and §7.2.3 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 §7.2.4.

We attempt to compare the properties of the self-supervised pretext tasks by replacing ODDS anomalies with synthetic variants in §7.2.5. Finally, we investigate the effect of architecture and detector choices in §7.2.6.

7.2.1 Self-supervision results

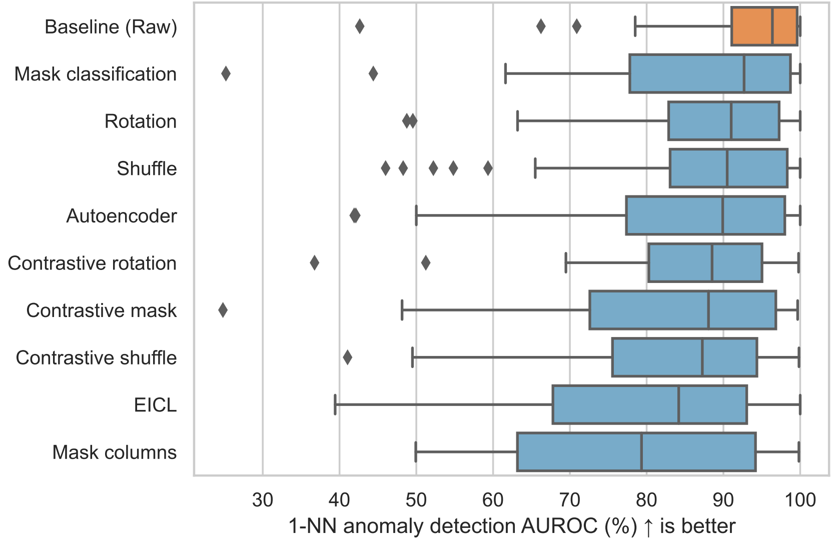


Figure 7.2: Box plot comparing nearest neighbour AUROCs for each of the embeddings, ordered by median performance. For each self-supervised task, we filter the results by architecture and loss function to include the embedding with the best-performing results.

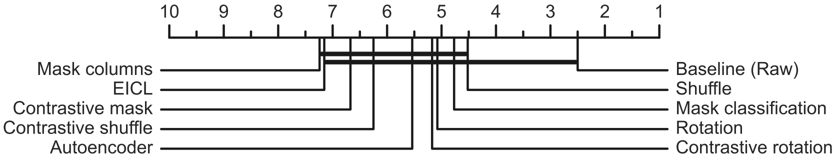


Figure 7.3: Critical difference diagram comparing the embeddings in a pairwise manner. The horizontal scale denotes the average rank of each embedding. The dark lines between different detectors indicate a statistical difference ($p < 0.05$) in results when running pairwise comparison tests. The baseline scores greatly outrank the pretext tasks. In contrast, the scores among the pretext tasks are more closely aligned.

No self-supervision task outperforms the baseline. Figure 7.2 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, Figure 7.3 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, Figure C.12). These results extend the findings in Grinszstajn et al. [273] 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.

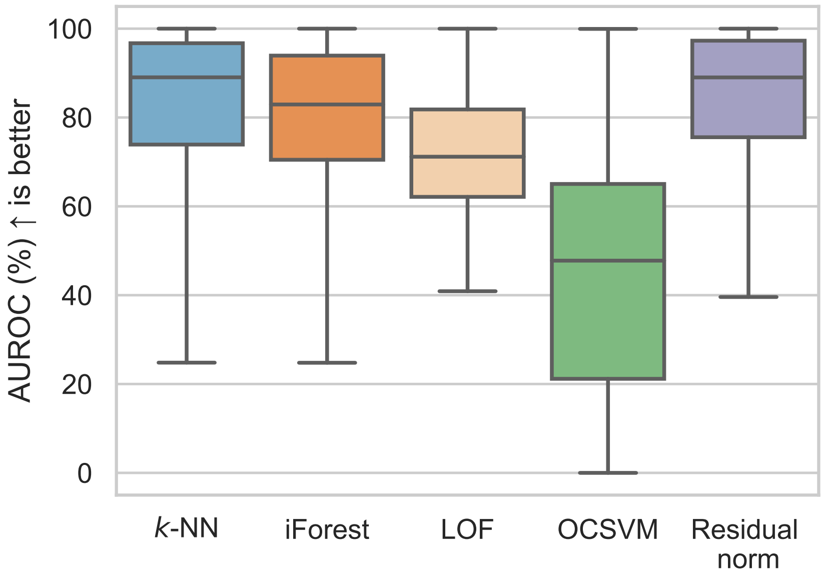


Figure 7.4: Box plot comparing detector performance on the self-supervised embeddings.

We observe similar results when we use different shallow detectors to perform anomaly detection (Figure 7.4), with one exception. Using residual norms on the embedding space is a slightly better choice than *k*-NN, as the interquartile range of scores is narrower. However, they still lag behind *k*-NN scores on the original embeddings. We also observe that OCSVM performs consistently worse across all tasks.

7.2.2 A case study on HTTP

To understand why self-supervision does not help, we 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 [280]. 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.

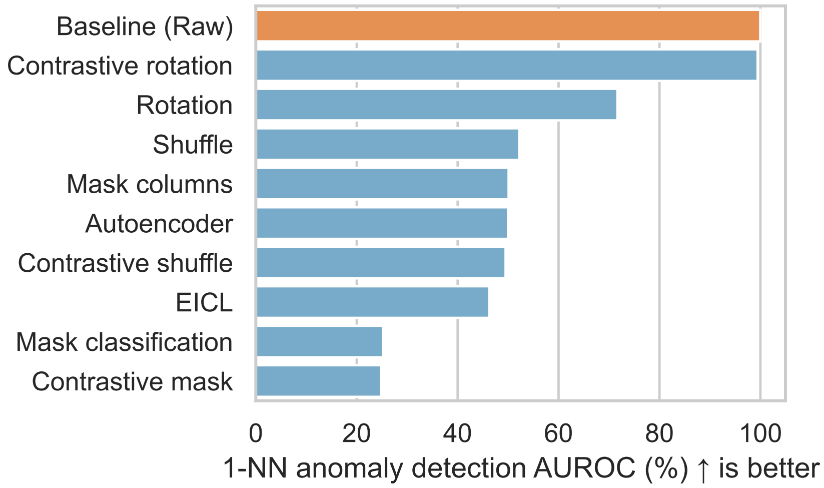


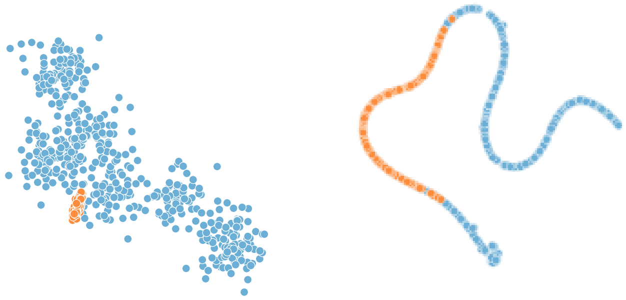
Figure 7.5: Bar chart comparing baseline and self-supervised embedding results on HTTP.

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 (Figure 7.5), sometimes yielding scores worse than random. Conversely, the other detectors maintain their performance. For example, when extracting features from the rotation task[1], 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.

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

7.2.3 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, Figure C.13). 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 fewer computational resources than the FT-Transformers. We apply the same architecture as the ODDS experiments, such that the extracted features are in a 128-dimensional space.



(a) Original 2D data

(b) t-SNE visualisation of the self-supervised features

Figure 7.6: Visualisations of the multiple Gaussian toy dataset. Light blue are the benign 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).

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 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 Figure 7.6.

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

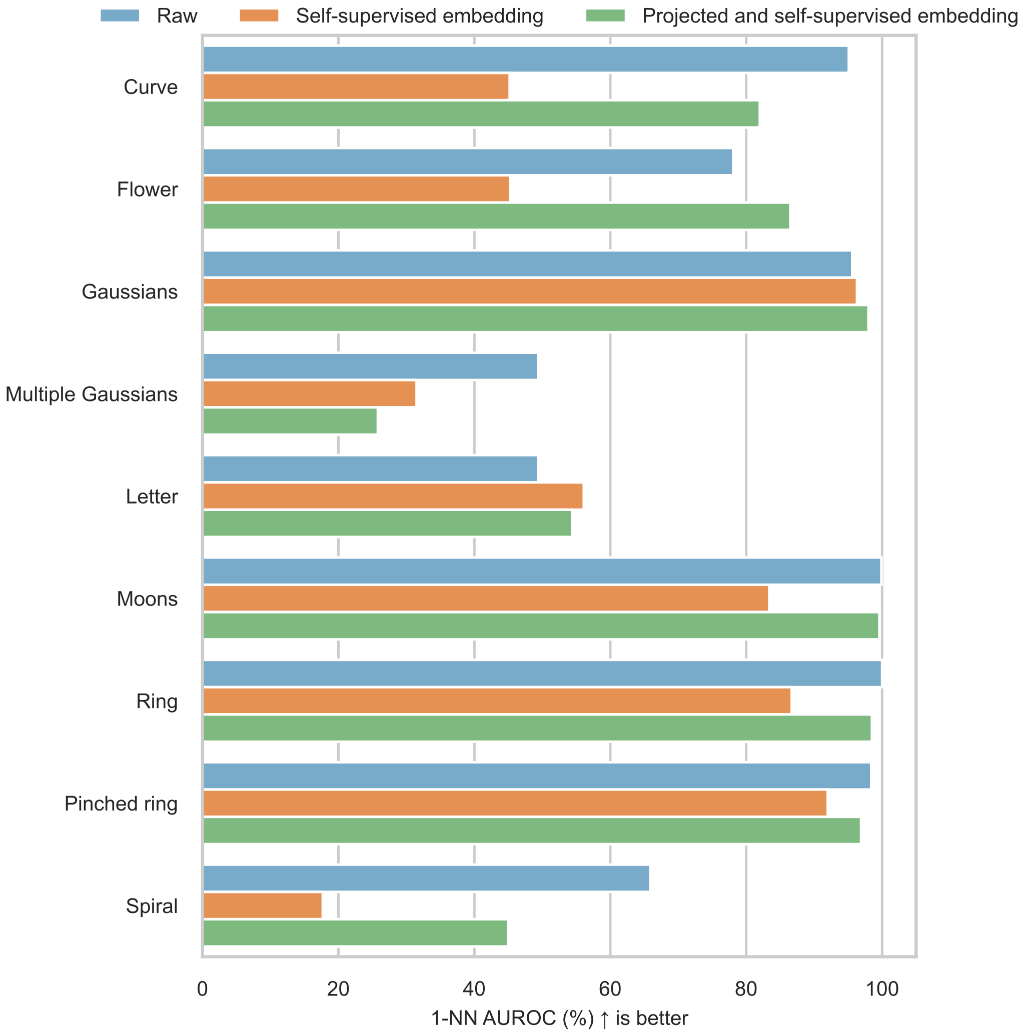
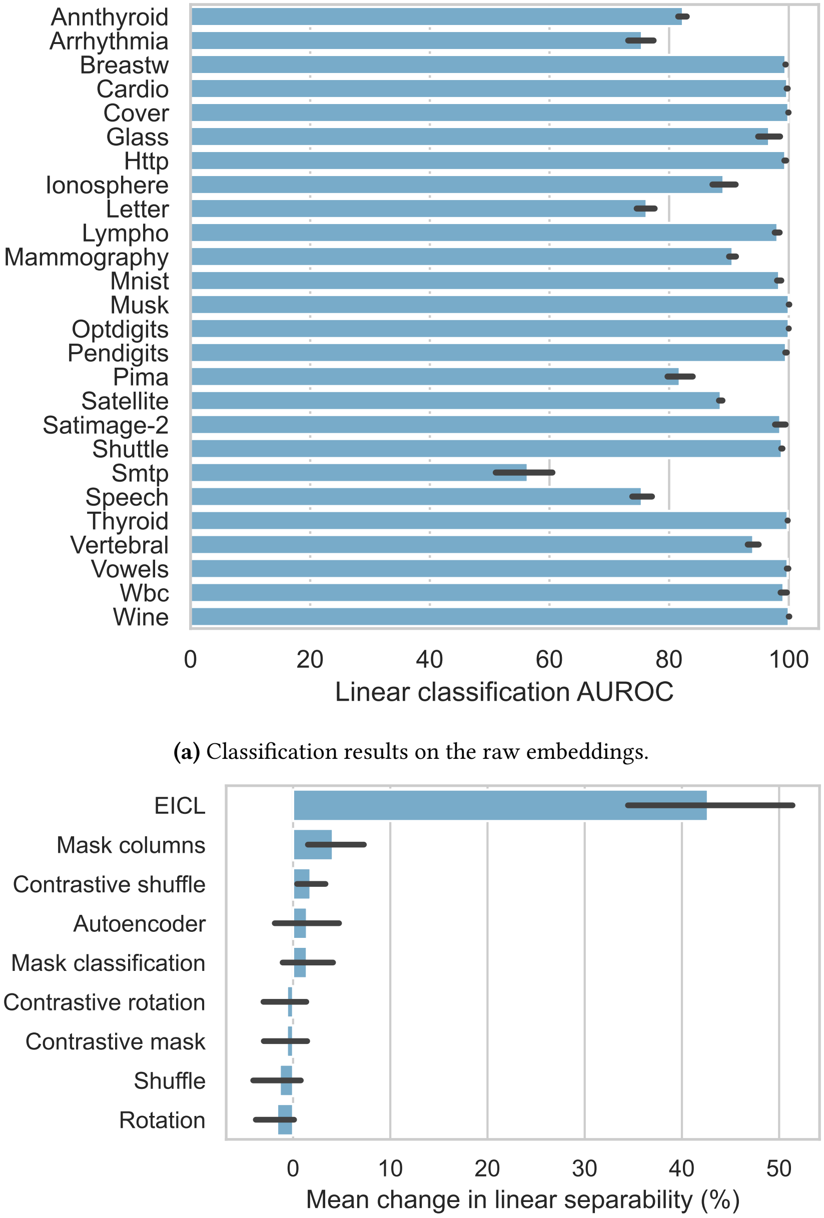


Figure 7.7: Nearest neighbour performance on the toy datasets. The raw embedding (blue) is the best in almost all instances. However, the self-supervision embeddings (orange) improve when projecting to a lower dimensional space (green).

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 (Figure 7.7), 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 benign 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.

7.2.4 Analysing ODDS embeddings



(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.

Figure 7.8: Supervised linear classification results (benign versus anomaly) on raw data (a) and supervised classification comparisons against the self-supervised embeddings (b).

We now proceed to run ablations on ODDS. Previous studies have shown that supervised classification performance correlates highly with out-of-distribution detection performance [257]. 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 with the benign samples. We could consequently attribute the poor self-supervised performance to this mixing rather than the presence of irrelevant directions.

Figure 7.8a shows 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 7.8b 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 benign data and anomalies. We can rule out the mixing effect and conclude that self-supervision generally does not affect the separability of the two classes.

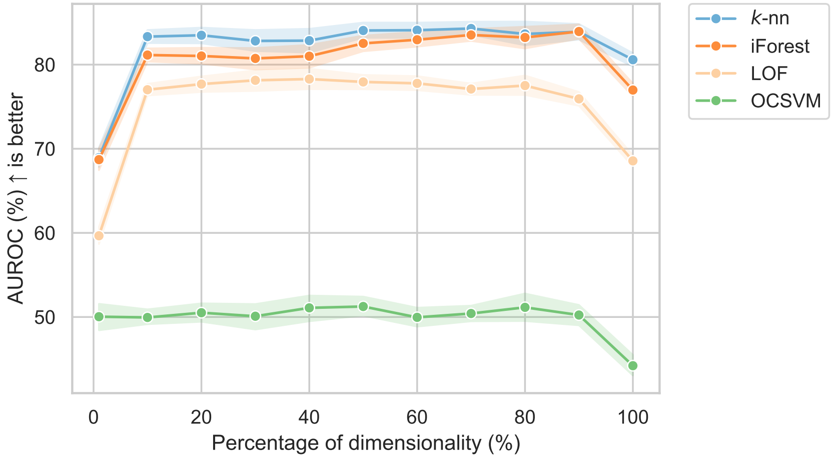


Figure 7.9: Ablation study showing how shallow detector results vary with subspace dimensionality, starting with the lowest eigenvalues.

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 7.9 shows the results. We aggregate both ResNet and FT-Transformer scores as we 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 Figure 7.9) results in lower AUROC scores than using a subset. Throwing away the top 10% of principal components garners most improvements, although performance generally remains stable when discarding more components - up to the top 90%.

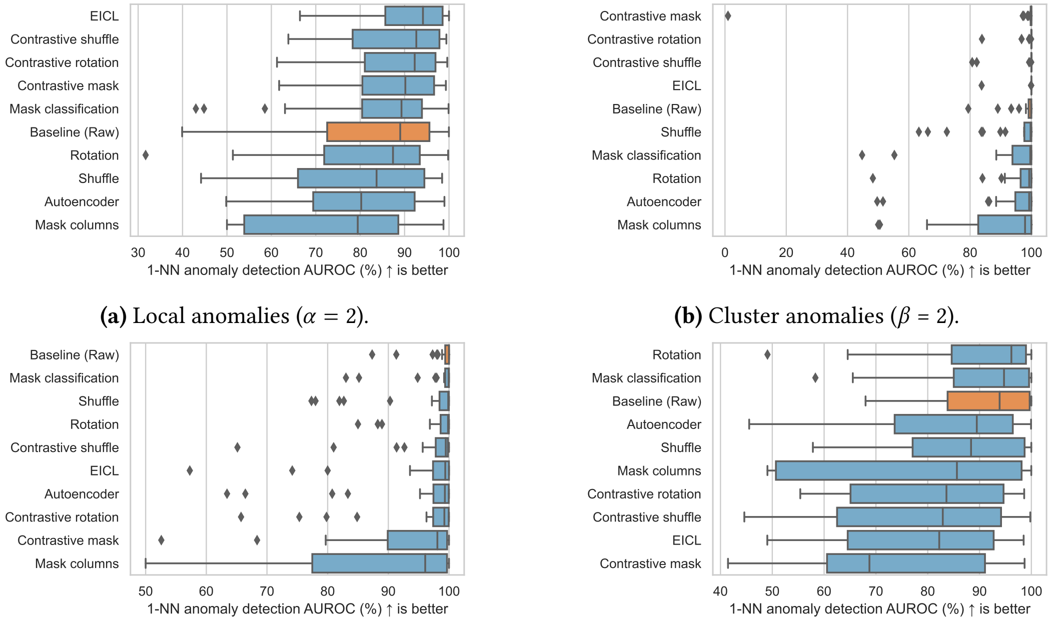
This observation aligns with previous findings that show residual directions capture information important for out-of-distribution detection [262, 281]. The magnitude of benign data is minute in this space which is not necessarily the case for anomalies. Based on these results, we do not need complete neural network representations to perform anomaly detection as a subset suffices.

7.2.5 Synthetic anomalies

Anomaly detection depends on two factors: the nature of the benign 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.

Figures 7.10a to 7.10d show 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 (Figure 7.10a) and cluster anomaly (Figure 7.10b) 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 (Figure 7.10c). 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 benign points. As a result, methods like k-NN become less effective. In addition, the ranking



(a) Local anomalies ($\alpha = 2$).

(b) Cluster anomalies ($\beta = 2$).

(c) Global anomalies ($\delta = 0.01$).

(d) Dependency anomalies.

Figure 7.10: Bar plots comparing synthetic anomaly results across the representations.

of the self-supervised tasks aligns most closely with their rankings on ODDS (Figure 7.13), which potentially highlights the overall properties of the ODDS datasets.

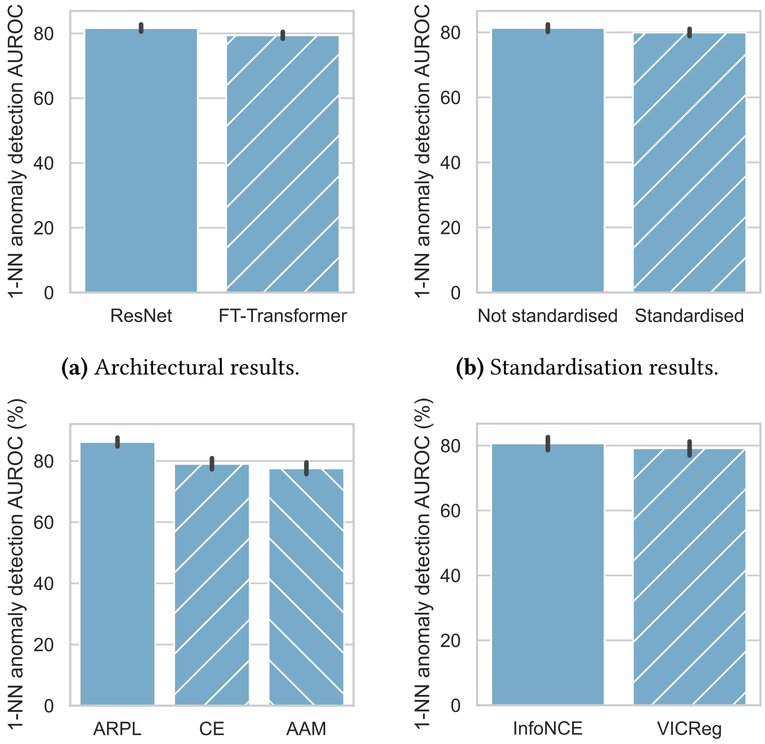
For the dependency anomalies, rotation and mask classification surpass the baseline (Figure 7.10d). Conversely, contrastive tasks perform the worst. Using a rotation or mask classification pretext task could help promote the intrinsic property that tabular data are non-invariant, which may help identify this type of anomaly.

7.2.6 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 that ResNets are a better choice than FT-Transformer (Figure 7.11a). This result may be due to transformers needing more training data during the learning phase [121] - the ODDS datasets are relatively small.

Standardisation is not necessary. Standardising data before training neural networks does not offer much benefit (Figure 7.11b). Due to the small size of the datasets, standardisation does not affect learning rates or performance.



(a) Architectural results.

(b) Standardisation results.

(c) Classification losses.

(d) Contrastive losses.

Figure 7.11: Comparisons of how architecture and losses affect performance on the self-supervised embeddings.

ARPL is a better choice for classification-type losses. ARPL significantly outperforms cross-entropy and AAM when training classification-type tasks (Figure 7.11c). 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 (Figure 7.11d) may be due to the intricacies of VICReg, which requires balancing three components (pair similarity, variance and covariance).

7.2.7 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 7.12 and 7.13 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

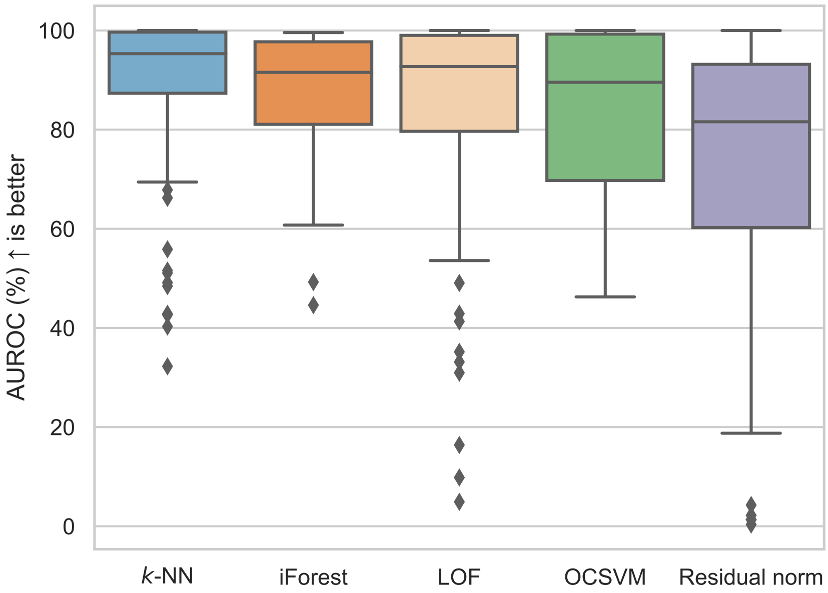


Figure 7.12: Box plot comparing detector performance on the raw and standardised data. The results include all hyperparameter variations where available.

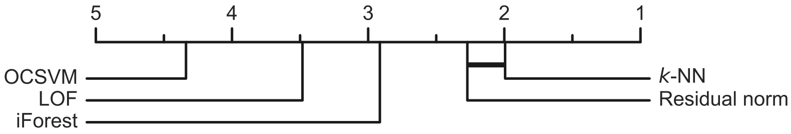


Figure 7.13: Critical difference diagram ranking the different detectors using Wilcoxon paired difference tests. The lines between different detectors indicate a statistical difference ($p < 0.05$) in results when running pairwise comparison tests.

[16, 67, 78]. However, apart from *k*-NN and residual norm, Figure 7.13 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 benign distribution. Nonetheless, the choice of underlying representation should take precedence over the detector when designing anomaly detection systems.

7.2.7.1 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 suboptimal self-supervised representations.

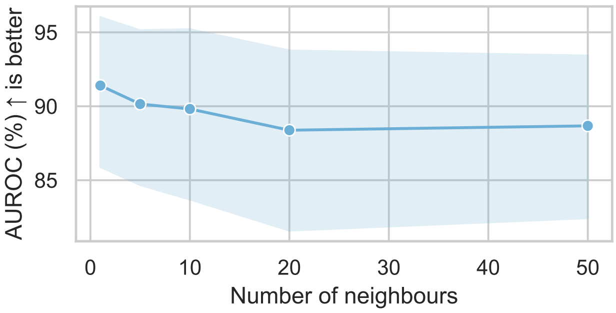


Figure 7.14: Line plot showing how k-NN varies with the change in the number of nearest neighbours, aggregated across the ODDS datasets, with 95% confidence intervals.

***k*-NN**: Figure 7.14 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 in ODDS already lie in distinct regions separate from the benign raw data.

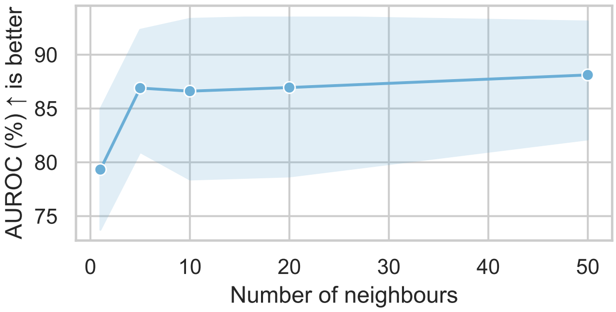


Figure 7.15: Line plot showing how LOF varies with the change in the number of nearest neighbours, aggregated across the ODDS dataset, with 95% confidence intervals.

LOF: Figure 7.15 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 7.16 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 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.

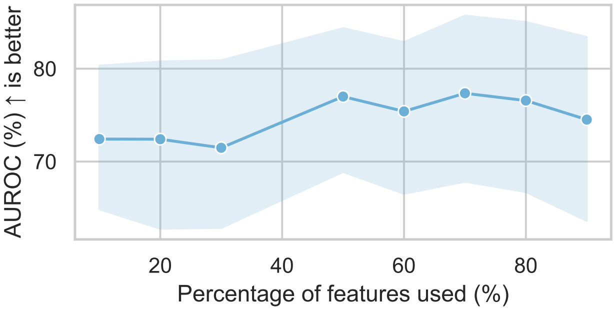
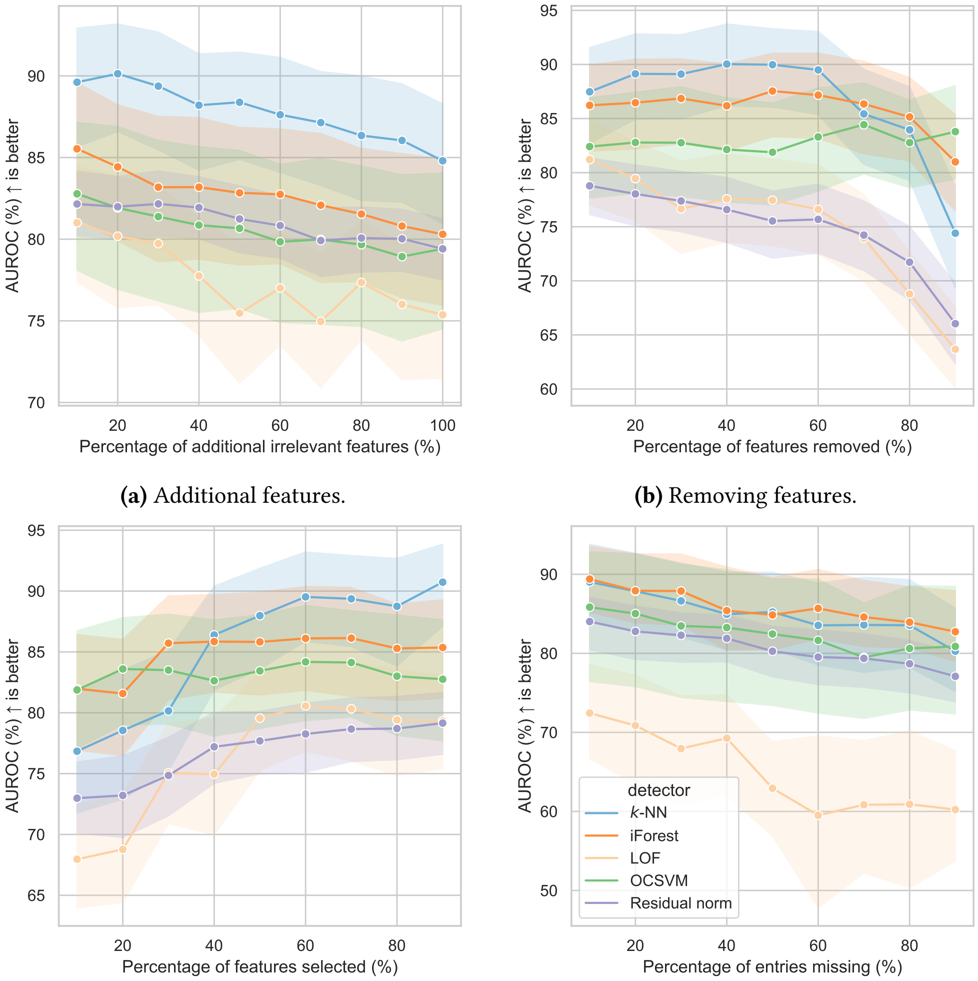


Figure 7.16: Line plot showing how residual norm varies with the change in residual dimensionality, aggregated across the ODDS dataset, with 95% confidence intervals.

7.2.7.2 Corrupted input data



(a) Additional features.

(b) Removing features.

(c) Selecting features.

(d) Missing values.

Figure 7.17: Ablations showing how detector performance varies with changing levels of corrupt data.

Adding uninformative features: All detectors are sensitive to irrelevant features (Figure

7.17a). 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 (Figures 7.17b and 7.17c).

Missing values: Most detectors exhibit a slight decline in AUROC with increasing proportions of missing values (Figure 7.17d). 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 presence of corrupted input data. As observed in our self-supervised results (§7.2.4), residual norms might be better at filtering out noisy directions. Furthermore, when there are fewer relevant features, iForest may be a better choice.

7.3 Conclusion

We conclude by outlining the limitations, directions for future work, and main contributions.

7.3.1 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 [121], 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 [39, 40]. 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 [16, 67, 78, 134], different parts of the neural network could provide more informative features [46]. Moreover, we chose to use shallow detectors to prioritise studying the effect of representations rather than studying the detection approach. 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 could yield better results [66].

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 [1, 15, 282]. 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 [1].

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.

7.3.2 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, confirming the observations in [15].

We conducted ablations to try to understand this behaviour. Our empirical findings suggested that neural networks introduce irrelevant features, which degrade detector capability. As benign 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 theories of why and when self-supervised learning works. Effective self-supervised pretext tasks learn to compress the input data when there are irrelevant features [283, 284, 285]. Our findings suggest current deep learning approaches do not improve performance when the original feature space succinctly represents the benign 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 benign 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.

8 | Conclusion

This thesis aimed to understand what types of representations enable one-class anomaly detection. To do so, we studied anomaly detection performance across different modalities. We conclude by summarising our findings, outlining strengths and weaknesses, and indicating areas of future research.

8.1 Contributions

We recap our findings for each modality covered in the thesis.

8.1.1 Images

We started with images (Chapter 3), as most anomaly and OOD detection research focuses on this modality [13, 17, 45, 72]. We presented a knowledge distillation framework to focus on what types of representations worked best for anomaly detection. We varied the underlying representation by altering the teacher model and training a student model to match the outputs of the teacher. We assessed anomalousness using the MSE between the student and teacher. This scoring method outperformed alternatives like Mahalanobis distance, suggesting that non-parametric detectors may be preferable. We evaluated our approach on an X-ray security dataset. Our method boosted the anomaly detection AUROC from 92.65% in a previous benchmark to 96.41% [17].

When analysing the representations, we found that teachers trained on semantic and rotation classification tasks worked best for semantic anomaly detection. Separability between benign and anomalous samples is insufficient for a reasonable representation. Instead, higher L2 gradient norms, which indicate more susceptibility to adversarial perturbations, are stronger signals for better representational candidates. For instance, we showed that SimCLR trained on CIFAR-10 underperformed on our anomaly detection tasks. Upon analysing the representation, it could reasonably separate samples in a supervised setting but had low L2 gradient norms.

8.1.2 Text

We then studied the effect of different representations for textual anomaly detection (Chapter 4). Starting from a pre-trained model [82, 83], we fixed the dataset and fine-tuned the models with various self-supervised objectives. We aimed to cover a range of self-supervision types by analysing masked language modelling [82], causal language modelling [129], and contrastive learning through SimCSE [135]. We used the self-supervision loss as the anomaly score.

Our approach worked better than other methods like bag-of-words, CVDD [116] and ELECTRA [130]. We demonstrated that these methods relied heavily on word frequency statistics through our word order anomaly experiments. Our fine-tuning approach set a straightforward baseline for future anomaly detection work. However, the findings suggest there is no clear-cut best self-supervision objective. The most appropriate choice depends on the type of anomaly. SimCSE is better at measuring differences in sentiment, causal language modelling is better at identifying discrepancies in word order, and masked language modelling lies between the two.

8.1.3 Speech

We pivoted to audio anomaly detection in Chapter 5, which is less explored compared to images and text. Due to the range of applications, we focused on speech deepfakes, which have already caused real-life harm [5, 6].

Chapter 5 created a benchmark for measuring how capable humans are at detecting speech deepfakes. We ran these experiments as we found a gap in the literature. Most speech deepfake detection work has concentrated on automatic speaker verification [163, 164, 165], with few studies looking at how humans perceive speech deepfakes. We asked 500 participants to listen to twenty clips and decide if they were spoken by a genuine human or by AI. No indication was given of the frequency of genuine or deepfake samples. We conducted the experiments in English and Mandarin and introduced a familiarisation treatment to see if humans could train to get better at detection. Our results suggest humans are unreliable, detecting deepfakes only 73% of the time. This proportion is not definitive and is likely optimistic, as the participants knew they would encounter deepfakes in the experiment. We also did not use state-of-the-art speech synthesisers to generate the deepfakes. Detection performance was similar between English and Mandarin participants, and familiarisation did not improve performance. However, we showed humans performed better than automated classifiers on OOD data. Our results highlight two directions for future research. As AI becomes more mainstream, a better understanding of how humans interact with AI is required. Secondly, future research should look at improving deepfake detection algorithms to generalise better to novel conditions. In the meantime, crowdsourcing human responses to assess deepfakes is a viable response.

Chapter 6 explores the viability of one-class detectors. Previous works on speech deepfake detection focus on binary classifiers [206, 207, 208], which generalise poorly to distribution shifts in both the bona fide and deepfake classes [14].

We analyse the performance of various pre-trained representations and the impact of different fine-tuning strategies. We show that one-class detectors can detect speech deepfakes, and general-purpose audio models offer the most benefit. Effective anomaly detection models occupy a low-dimensional subspace. Deep representations identify TTS deepfakes more consistently than VC deepfakes. This issue persists across deep representations.

Moreover, our results suggest that one-class classifiers are more robust to bona fide distribution shifts than supervised classifiers. Overall, our analysis highlights the value of one-class methods for speech deepfake detection.

8.1.4 Tabular data

Our final Chapter (Chapter 7) investigates why deep learning representations do not improve tabular anomaly detection. The experiments build on a previous proof-of-concept that showed raw features outperformed specialist deep tabular anomaly detectors [15]. We trained shallow anomaly detectors with representations formed from appropriate tabular pretext tasks. None of these representations were better than the raw baseline. Further ablation studies suggested that the neural networks introduced irrelevant features. However, we demonstrated performance was recoverable by projecting the embeddings to a residual subspace.

We also provided insights that may help practitioners wishing to implement anomaly detectors. Although the representation is the most crucial aspect of anomaly detection, localised detectors like k-NN outperformed the other detectors. However, these localised detectors were susceptible to performance degradation on corrupted data. In these situations, isolation forests may be preferable.

In addition, we indicated there might be specific scenarios where self-supervised pretext tasks may be beneficial. Contrastive tasks were better at finding localised anomalies, while classification tasks were better at pinpointing changes in dependency structures.

8.2 Summary

Our analyses across modalities suggest there is no panacea for anomaly detection. The choice of representation affects performance more than the detector. Deep representation learning methods only aid anomaly detection if the mapping removes redundant information and captures the core semantic features in the benign data distributions. Learning such a mapping is generally implausible for tabular data as we do not know what the inductive biases are. In contrast, regularities in images, text, and audio means there are straightforward pretext tasks we can leverage.

An appropriate representation depends on the characteristics of the benign data and the anomalies. Using embeddings from a pre-trained model is a sensible starting point. Models trained on mass amounts of data should incorporate prior information about both classes. Our results suggest that models trained on a similar domain are preferable but should not be too task-specific. Highly specialised models would not contain sufficient discriminative features to identify anomalies. For example, the speech experiments in Chapter 6 show speech and audio networks are better than image-based networks. However, a more general-purpose audio network encodes anomalous features more explicitly than a speech network.

Pre-trained representations offer the most benefit, although fine-tuning can give a slight performance boost. The choice of fine-tuning objective depends again on the benign data and anomalies. The centre loss is a domain-neutral choice but is prone to representational collapse and is less effective for multimodal distributions. Constructing a classification task using metadata or perturbations can aid more fine-grained anomalous variants, whereas contrastive objectives are better for clustered anomalies. Overall, a reasonable rule of thumb for anomaly detection across modalities is fine-tuning a pre-trained network trained on a diverse dataset with benign data.

There is no consistent metric for measuring the quality of representations. Our results in images and text suggest separability between anomalies and benign data is insufficient for good anomaly detection. Measures like L2 gradient norms and eigenspectrum alphas might provide some hints but are architecture and dataset-specific, making large-scale comparisons challenging. This issue is addressable by incorporating prior knowledge of anomalies, reducing the potential evaluation space. The choice of metric should be task-dependent. Perhaps evaluating the detectors on carefully curated validation sets or human comparisons would be the most explainable and relevant way to measure representation quality.

When analysing the practical aspects of anomaly detection, we find no detector is significantly better than another. Non-parametric methods like nearest neighbour distance work slightly better than the others but decline in the presence of corrupted features. Our initial experiments suggest ensemble methods like isolation forest may be more robust, but further investigation is required. Introducing a sense of "otherness" is beneficial for the pretext tasks. We can include a concept of otherness through the data or the objective. One could create synthetic anomalies through perturbations or external datasets, although this relies on knowledge of the anomalies. Regarding objectives, cross-entropy loss performs reasonably well for classification-based tasks, but ARPL might be better for small data regimes.

8.2.1 Ethical considerations

Security measures often include anomaly detectors. The thesis highlights examples such as deepfake detection. Although detectors can help detect suspicious activity, anomaly detectors in high-stakes settings could negatively impact people. For example, there are reports that AI text detectors are unfairly biased against non-native speakers [286], and biometric systems often perform poorly on deeper skin tones [287].

Solely automated decisions with discriminatory outcomes could breach UK laws [288]. Developers should test their detectors on benign edge cases and apply safeguards to mitigate harm. For example, they should ensure there are ways for people to challenge a detector and request meaningful human review.

Malicious actors could use anomaly detectors to improve their attacks. For instance, they could use a detector's outputs to refine their deepfake generation pipeline and evade future defences. The cat-and-mouse dilemma is a perpetual issue in security. However, open research on defences benefits the larger security community. Developers can also restrict access to their defences, such as limiting white-box access and only providing binary pass or fail results.

8.2.2 Limitations

We conducted our anomaly detection experiments in settings that do not necessarily reflect realistic deployment environments. One aspect is the choice of datasets. The datasets that we analysed are specifically designed for academic research. Hence, they are clean and carefully labelled. Although we ran some preliminary experiments to see how anomaly detection changed with noisy speech (Chapter 5) and synthetically corrupted tabular data (Chapter 7), more experiments with corrupted data (such as missing features or contamination) would be beneficial.

We also primarily analyse semantic anomalies. In this setting, samples are either purely benign or completely unusual. In practice, the situation may be more unclear. Adversaries might conceal anomalies by mixing them with genuine content. For example, partial speech deepfakes [289] contain a mixture of bona fide and synthesised utterances. Future work could extend our anomaly detection studies to partial anomalies or look at how to alter decision functions to take partial anomalies into account.

We investigate various techniques for measuring the quality of the representations. Although we decided to incorporate new representation learning techniques, our results would be more comparable if we used the same approaches across modalities. Measuring representations using only the training data may not illustrate the actual representation space. We show that anomaly detection performance depends on the nature of benign data and anomalies. We have acknowledged this drawback in the Background section (§2.2.2). In practice, incorporating a wide range of measures, acknowledging their pros and cons, and analysing performance across all subsets of data might be better.

Other aspects beyond representation also contribute to the choice of anomaly detector. A detector might be precise but not usable if it is too slow. Alternatively, we should consider the decision-making process after observing anomaly detector outputs. Do we rely on the detector to make the decisions, or do they require human oversight? Moreover, the interpretability of a detector's outputs could also influence the utility of a detector. These concerns are outside of the scope of the thesis but are worth considering when deploying anomaly detection models.

8.2.3 Future work

Our results show that fine-tuning a pre-trained network is a reasonable benchmark for anomaly detection. When running this benchmark in practice, future work should look at a detector's interpretability and resilience to corrupted inputs. These aspects are crucial because the data pipeline is not always as clean as the academic setting. Starting points could include comparing anomaly detector outputs to human decisions (per our speech deepfake experiments in §5.1), studying performance changes under varying degrees of data corruption, or evaluating more fine-grained anomalies.

Another issue with one-class learning is that there are multiple ways in which OOD samples can manifest, especially in high-dimensional spaces like images. Including prior information about anomalies narrows this problem space. Although collecting genuine anomalies might be challenging, future work could use synthetic proxies instead [154].

Moreover, we examine anomaly detection performance on one modality at a time. Recent research has pivoted to multimodal models [290, 291]. Additional work could study how anomaly detectors perform in a multimodal setting to identify overarching trends.

An alternative approach that complements anomaly detection is to flip the problem statement. Instead of searching for unusual instances, we could probe for signs that an instance is benign. In some situations, trying to search for signs of unusualness is an impossible task. We might have a better understanding of credibility. The C2PA provenance standards are one initiative adopting this angle [292]. One can establish an asset's source and edits by looking at its metadata. Although C2PA primarily aims to combat disinformation, we could potentially build machine learning detection models with similar decision-making rationale. Taking X-ray baggage imagery, we could incorporate features like what goes into a typical traveller's luggage. Provenance could be a sensible complement to anomaly detection.

Overall, our experiments show that anomaly detectors have utility. They generalise better to unseen anomalies than binary classifiers. Nonetheless, anomaly detection is only one part of safety. Ultimately, we should look at how to include anomaly detectors in a more holistic decision-making pipeline. As technology and AI become more central to our lives, studies beyond the scope of representation learning are necessary. Building on the impact of our work that shows humans cannot reliably detect deepfakes, further studies on how anomaly detectors can complement human capabilities would be insightful.

Appendices

A | Out-of-distribution sample detectors

OOD sample detectors are either *shallow* or *deep*.

Shallow approaches do not perform any representation learning. They only classify points and rely on fixed feature maps. Traditionally, hand-crafted feature engineering (e.g., mel spectrograms for audio [24]) produced feature maps. Embeddings from deep neural networks have replaced hand-crafted features. Neural network embeddings require less hyperparameter tuning and can better exploit the intrinsic properties of data [26, 293]. However, the two-stage process of shallow detectors may struggle to scale to large and complex datasets.

In contrast, deep variants simultaneously transform data to a new subspace and perform classification. The end-to-end setup eliminates the need for manual feature engineering and automatically can learn hierarchical representations for data. Nevertheless, deep methods come with disadvantages. They are more data-hungry [294] and are less interpretable than shallow methods [42].

The choice of detector depends on whether its assumptions are more appropriate than another for the downstream task. We summarise the methods that are used in this thesis below. Ruff et al. [1] and Salehi et al. [22] contain a more detailed overview of the various approaches.

A.1 One-class detectors

Beyond the shallow versus deep categorisation, detectors can be categorised by how they assess anomalies. Broadly, these decisions are classification-based, probabilistic, use reconstructions, or use distances.

Figure summarising one-class approaches.

Figure A.1: Graph depicting one-class approaches spanned by model type and depth (from Ruff et al. [1]).

Classification models learn a decision boundary that separates the data into two classes (benign or anomalous) according to some labelling regime. This arrangement is more complicated in the one-class setting as no labelled anomalous training samples are available. One-class models address this using proxy anomalies, such as the origin.

Probabilistic approaches use generative modelling to approximate the benign data distribution. Points lying in low-density regions are deemed more anomalous.

Reconstruction methods, also known as **dictionary** methods, assume the building blocks of a feature space can reconstruct benign data but cannot construct anomalies. Methods using dictionaries use either linear or non-linear manifold learning techniques (e.g., principal components analysis or autoencoders) to determine the building blocks [46, 262, 295].

Distance-based methods differ from the previous approaches as they do not involve an explicit a priori training phase [1]. Evaluations on test points occur in an online fashion. Each distance-based method has a predefined way of measuring distance. Given this distance and a set threshold, if the test point lies far from the training data, it is deemed more anomalous.

The below sections and Figure A.1 highlight shallow and deep methods from the above four categories. Generally, methods for anomaly and novelty detection are interchangeable.

A.1.1 Shallow methods

One-class support vector machine (OCSVM) assumes benign data lies in a high-density region [282]. Taking the origin as an anchor in the absence of anomalous training data, 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. Given a dataset $\mathcal{D} = \{x_1, ..., x_n\}$, the objective function for OCSVM is as follows:

$$\min_{\boldsymbol{w},\rho,\boldsymbol{\xi}} \quad \frac{1}{2}||\boldsymbol{w}||^2 - \rho + \frac{1}{\nu n} + \sum_{i=1}^{n} \xi_i \tag{A.1}$$

$$\text{s.t.} \quad (\boldsymbol{w} \cdot \theta(\boldsymbol{x}_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad \forall i \tag{A.2}$$

Where $\boldsymbol{w}$ is the maximum margin hyperplane, ρ is the distance from the origin to the hyperplane and $\boldsymbol{\xi}$ are the slack variables. $||\boldsymbol{w}||^2$ is a regulariser, ν is a hyperparameter that controls the trade-off between the two terms, and $\theta(\boldsymbol{x}_i)$ is a kernel transformation of the data points.

Support vector data description (SVDD) closely relates to OCSVM [65]. Instead of finding a maximum margin hyperplane to separate the two classes, it learns the smallest hypersphere to capture most of the benign data. Any points that lie outside of the hypersphere are considered anomalies. SVDD is equivalent to OCSVM when using Gaussian kernels. The objective function is as follows:

$$\min_{R,\boldsymbol{c}\boldsymbol{\xi}} \quad R^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i \tag{A.3}$$

$$\text{s.t.} \quad ||\theta(\boldsymbol{x}_i) - \boldsymbol{c}||^2 \leq R^2, \quad \xi_i \geq 0, \quad \forall i \tag{A.4}$$

Where $R > 0$ is a radius, $\boldsymbol{c}$ is a centre in the feature space, $\boldsymbol{\xi}$ are the slack variables and ν is a trade-off parameter. The scoring function is therefore $||\theta(\boldsymbol{x}_i) - \boldsymbol{c}||^2$. If it is greater than R^2, the point is anomalous.

The **Mahalanobis** distance measures the distance between a point and a distribution. It assumes the data follows a normal distribution. Any point exceeding a given distance threshold is considered unusual. The score is as follows:

$$\sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^T \cdot \Sigma^{-1} \cdot (\boldsymbol{x} - \boldsymbol{\mu})} \tag{A.5}$$

Where $\boldsymbol{x}$ is the point that is evaluated, $\boldsymbol{\mu}$ is the mean of the distribution and Σ is the covariance matrix of the distribution.

Gaussian mixture models (GMM) are a type of probabilistic model. It assumes the underlying distribution of the benign data is a mixture of Gaussians. In the one-class anomaly detection setting, a GMM is fit on the benign data. Points lying in low-density regions are considered more unusual.

In addition to the distributional assumption, the effectiveness of GMMs relies on tuning multiple parameters. These parameters include the number of components, the mixing weights between each component (the probability that a datum belongs to one component), and the initialisation methods for the component.

Principal components analysis (PCA) is a linear dimensionality reduction method. It transforms the data into a new coordinate system where most data is characterisable using fewer dimensions than the original input dimensionality. The principal components, which describe the primary variance directions, are the new coordinate system. The eigenvectors of the data's covariance matrix denote the principal components.

Anomaly detectors using PCA assume the principal components can describe benign points but fail to depict anomalies. The scoring function is the reconstruction error between the original data and its reconstruction using PCA [295].

Residual norms are closely related to PCA. They have achieved state-of-the-art performance for out-of-distribution detection on images [262]. 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 K largest eigenvalues of $\mathbf{X}^T\mathbf{X}$. We assume anomalies have more variance on the components with smaller explained variance [281]. Therefore, we project $\mathbf{X}$ to the subspace spanned by the *smallest* eigenvalues of K (represented by $\mathbf{W}^{\perp}$) to encapsulate the residual space. We take its norm as the anomaly score:

$$||\mathbf{x}^{W^{\perp}}|| \tag{A.6}$$

***k*-nearest neighbours** (k-NN) assumes benign 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 [15, 16, 67, 78, 79]. k-NN typically uses features extracted from pre-trained classification neural networks [16, 67, 79] for anomaly detection in domains where such networks are available, such as images.

Local outlier factor (LOF) is a density-based outlier detection method [296]. 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 [261]. 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.

A.1.2 Deep methods

Deep SVDD (DSVDD) is similar to SVDD. In addition to the smallest hypersphere, it learns a new mapping for the data using an autoencoder [66]. It is faster and performs better features than shallow SVDD, which relies on quadratic-scaling kernel functions to transform input data [297]. The objective function is as follows:

$$\min_{\mathcal{W}} \frac{1}{n} \sum_{i=1}^{N} ||\theta(\boldsymbol{x}_i; \mathcal{W}) - \boldsymbol{c}||^2 + \frac{\lambda}{2} \sum_{=1}^{L} ||\mathbf{W}||^2 \tag{A.7}$$

The first term is the centre loss, where $\theta(\boldsymbol{x}_i; \mathcal{W})$ is the neural mapping of $\boldsymbol{x}_i$. The second term is a network weight decay regulariser with hyperparameter $\lambda > 0$.

Unlike shallow SVDD, DSVDD does not explicitly penalise the radius. The authors state the same effect is achievable through neural mapping. However, DSVDD requires engineering tricks to avoid feature collapse. Examples include fixing the centre representation instead of learning it, removing bias terms from the neural network architecture and using unbounded activation functions.

Pre-trained anomaly detection adaptation (PANDA) [67] also uses a centre loss like DSVDD. However, it uses a pre-trained network to extract features. The method uses early stopping and elastic weight regularisation to prevent feature collapse.

Autoencoders are a type of neural network that learns to recreate input data. They are capable of performing non-linear dimensionality reduction. Sometimes, modifications to autoencoders (e.g., adding a denoising aspect [102] or using variational inference to train [298]) can improve the learnt representation. Like PCA, the anomaly scorer of choice is the input reconstruction error [71]. Improvements to autoencoder-based anomaly detection incorporate the latent reconstruction error [45, 46].

Geometric-transformation classification (GEOM) [80] and similar methods (Rotation prediction [113] or GOAD [23]) transform the one-class configuration to an open-set problem using perturbations. These methods transform the benign data subspace $\mathbf{X}$ into K subspaces $\mathbf{X_1}, ..., \mathbf{X_K}$ using augmentations (e.g., rotations, reflections, translations). A neural network learns to predict the perturbation applied to the data. The overarching idea is that a trained neural network can correctly predict benign data augmentations but cannot do so on anomalies. Therefore, the anomaly score is typically the classification confidence or the loss.

A.2 Multiple class detectors

Detectors in this category rely on the training set subclasses to perform inference.

Mahalanobis approaches are extendable to multiple classes. Instead of comparing a test datum's distance to the mean and covariance of the entire training set, the computation compares its distance to the mean and covariance of each subclass [96]. We take the smallest distance between the datum and a subclass as the anomaly score. Extensions of this approach divide this score using the background value, which is the original one-class Mahalanobis distance [299].

Maximum softmax methods use the intuition that classification networks are less confident on OOD samples. Therefore, the highest confidence from the softmax layer of a network serves as the score [13]. Extensions to this work involve adversarially perturbing the inputs to improve in-distribution and OOD separability [300] or using the values from the logits layer [301].

B | Neural network architectures

In this section, we outline the different neural network architectures used in the thesis.

B.1 Inductive biases of different architectures

Neural networks learn to map input data to a specified output. Different architectures leverage different inductive biases. These biases guide the model to focus on particular elements of the data. These assumptions help adapt anomaly detection pipelines to varying modalities. The architectures that primarily feature in the thesis are as follows:

B.1.1 Convolutional neural networks

Convolutional neural networks (CNNs) specialise in processing grid-like data. They use filters (also known as kernels) to learn relevant features. These filters, optimised through training, slide across the input data to extract localised patterns. This setup also enables CNNs to be spatially invariant, meaning that they are well-suited for tasks where the location of relevant features is unimportant [101]. Several studies also suggest CNNs learn hierarchical representations [101]. The earlier layers of a CNN recognise more rudimentary edges and textures, while the later layers focus on more complex objects, such as semantic categories [101]. CNNs are a popular option for processing images [26] and two-dimensional audio representations, such as spectrograms [302].

B.1.2 Transformers

Transformers specialise in learning context across sequential data. Initially proposed for text [127], they have quickly become the architecture of choice for other modalities like images and audio [121, 249]. The core aspect of transformers is the self-attention mechanism [127]. Self-attention allows a model to determine the importance of different parts of an input sequence and weight these based on the output. Transformers across varying tasks and modalities have the following commonalities:

Embedding: A tokeniser splits the input sequence into smaller units called tokens. The tokeniser also maintains a dictionary of possible tokens. Using the dictionary, it maps the tokens into numerical token IDs so the model can process them. The tokeniser also ensures all inputs are the same length by truncating longer sequences or padding shorter ones.

Additional steps are needed to process image and audio data, which are not discrete. The first step in processing images is to divide the image into non-overlapping patches. The transformer pipeline projects the image to a lower dimensional space and flattens it to a one-dimensional sequence [121]. For audio data, the same steps apply to its spectrogram representation [249]. Alternatively, a feature encoder learns to map the waveforms into discrete components [216, 217, 218].

Positional encoding: As the self-attention mechanism processes entire sequences, transformers need additional guidance to capture positional information. The data processing procedure adds fixed positional encodings to input sequences. These embeddings depict the token position in text and the row and column position of the patch in images.

Attention: Transformers learn three weight matrices in each self-attention block: the query weights $\mathbf{W_Q}$, the key weights $\mathbf{W_K}$, and the value weights $\mathbf{W_V}$. The queries are elements we want to learn contextual information about. For an input token $\mathbf{x_i}$, the query vector is $\mathbf{q_i} = \mathbf{x_i}\mathbf{W_q}$. We learn these relationships from keys. These keys are the same as the queries in self-attention, hence for $\mathbf{x_i}$, its corresponding key is $\mathbf{k_i} = \mathbf{x_i}\mathbf{W_k}$. Separate $\mathbf{W_Q}$ and $\mathbf{W_K}$ matrices mean that attention does not need to be symmetric. If token i gives relevant context to token j, the reverse does not need to apply. The values contain the actual information associated with each element. So for a datum $\mathbf{x_i}$, this is $\mathbf{v_i} = \mathbf{x_i}\mathbf{W_v}$.

The dot product between the queries and keys is the attention score. The score is normalised and scaled with a softmax and scaling factor ($\sqrt{d_k}$) to ensure numerical stability during training. Finally, the normalised attention scores multiplied by the value matrix produce the final context vectors. In matrix notation ($\mathbf{Q} = \mathbf{X} \cdot \mathbf{W_Q}, \mathbf{K} = \mathbf{X} \cdot \mathbf{W_K}, \mathbf{V} = \mathbf{X} \cdot \mathbf{W_V}$), attention is as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{B.1}$$

Self-attention is parallelisable and is faster than sequential architectures like recurrent neural networks.

Multiple heads: Transformers use multiple attention heads to capture different relationships. For example, in text, one could attend to the next word in a sequence while another could attend to nouns.

B.2 Overview of neural networks used

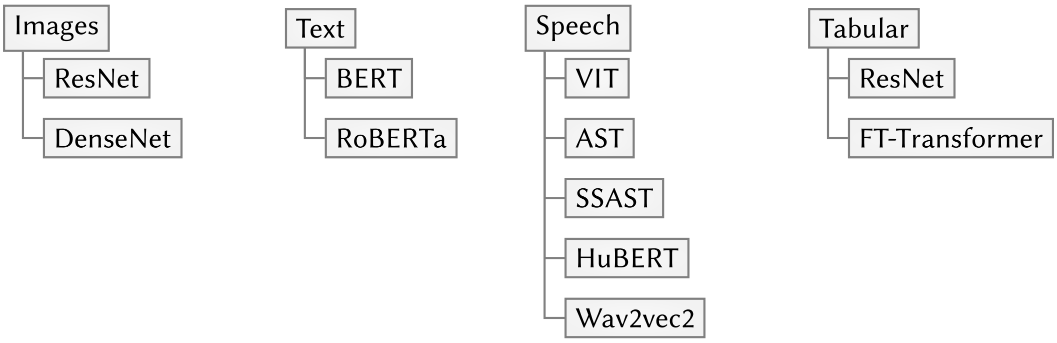


Figure B.1: Tree diagram of neural networks used by modality.

ResNet is a CNN architecture that introduced the concept of skip connections to resolve the vanishing gradient problem [276]. Deeper neural networks perform better because they learn more complex hierarchical features [101]. The backpropagation process means neural network weights update using a proportion of the gradient of the loss function. The gradient updates can be close to zero when backpropagating from the output layer to the input layer. As a result, these "vanishing gradients" can hinder training [303]. ResNets use residual blocks to mitigate this problem and to enable deeper networks. A residual block is a series of convolutional layers with skip connections. Namely, if the input is $\boldsymbol{e}$, the output would be $F(\boldsymbol{e}) + \boldsymbol{e}$, where $F(\boldsymbol{e})$ is the output of the convolutional layers. ResNets have benefitted tasks beyond image classification, including audio [302] and tabular data classification [275].

DenseNet is another CNN architecture that addresses the vanishing gradient problem [112]. It uses dense connections instead of skip connections. The inputs of each convolutional layer comprise the output from the previous layer and all other preceding layers in the block. Namely, if the block has l layers, the l-th layer's output $\boldsymbol{e}_l$ is the concatenation of the preceding feature maps: $x_l = F([\boldsymbol{e}_0, \boldsymbol{e}_1, ..., \boldsymbol{e}_{l-1}])$.

BERT, or "bidirectional encoder representations from transformers", is an NLP model introduced in 2018 [82]. BERT uses a conditional prediction task to learn features. In particular, it uses a masked language modelling objective, which enables the model to learn contextual information in a bidirectional manner. The original version of BERT masks 15% of tokens with either a `[MASK]` token or a random word. The model learns to decipher the correct word using a fixed vocabulary set. BERT also trains with a next-sentence prediction task. This task predicts whether a given sentence follows another one in the text. However, follow-up work suggests the BERT's main benefit comes from the masked language modelling task [83].

RoBERTa improves on BERT's training process [83]. The model trains for longer and with more data. The authors also remove the next-sentence prediction task and alter the masking scheme. BERT only generated masks during the pre-processing stage. This setup means the model sees repeated masked sequences during training. RoBERTa addresses this by producing masked sequences on-the-fly to encourage better representations.

Vision transformers (VIT) adapt the original transformer model for computer vision [121]. They do so by splitting the image into non-overlapping patches. The patch size is a hyperparameter that depends on the downstream task. A feature encoder (similar to a convolution layer) linearly projects and flattens the patches into lower-dimensional vectors.

Audio spectrogram transformer (AST) adapts VIT to audio using two-dimensional spectrograms as input [249]. The original AST uses pre-trained VIT weights as a base, as transformers need more data to train than CNNs [121].

Self-supervised audio spectrogram transformer (SSAST) uses AST as the underlying architecture. However, it differs from AST as it uses self-supervised learning [253]. The task, "masked spectrogram patch modelling", combines conditional prediction and perturbation classification. The model receives patches of a two-dimensional spectrogram as input. A learnable mask embedding replaces a portion of the patches randomly. The model learns to find the correct patch from all masked patches and reconstruct it.

Wav2vec 2.0 is a self-supervised speech representation model [218]. Wav2vec 2.0 also uses a mask classification pretext task. Its feature encoding module processes raw waveforms into learnt discrete units using a convolutional feature encoder and a quantisation module. After feature encoding, a learnable mask embedding replaces some patches randomly. Wav2vec 2.0 looks at 100 patches for each masked position and identifies the original unperturbed patch. The remaining patches are negative distractors. These distractors come from other locations in the same sequence. It differs from Wav2vec 1.0 [217] as it uses a transformer instead of a CNN for the primary representation learning architecture.

Hidden-unit BERT (HuBERT) is another self-supervised speech representation model [216]. HuBERT uses an alternating two-step training process. The first stage learns the discrete speech units. This process involves converting the raw audio waveform into latent features and k-means clustering. At initialisation, the initial features are mel-frequency cepstrum coefficients [304]. At later stages, the model reuses the latent parts from an intermediate layer of HuBERT's transformer encoder. The second step adapts BERT's masked language modelling task [82]. A learnable mask vector replaces 50% of the latent feature vectors, and the model predicts the correct units. The prediction logits use the cosine similarity between the transformer outputs and the hidden embeddings from the first step. The loss only uses the masked positions. HuBERT's loss objective differs from Wav2vec 2.0 as it uses cross-entropy loss instead of a contrastive loss.

Feature tokenizer + Transformer (FT-Transformer) adapts transformers to tabular data [275]. The feature tokeniser model transforms numerical and categorical features to embeddings and applies a stack of transformer layers to the embedding. Therefore, every transformer layer operates on the feature level of one datum.

C | Supplementary results

C.1 Images

Table C.1: Anomaly detection results for the "cat" class in the Cats vs. Dogs dataset.

		Anomaly Detection Method (AUROC)			
Teacher Representation	**Classification Acc.**	Knowledge Distillation	MSE	Mahalanobis (Diagonal)	Mahalanobis (Full)
Baseline	*83.40*	*88.67*	*88.51*	*75.70*	*85.11*
Random	64.11	48.46	52.50	52.03	53.52
STL Classification	71.28	58.99	60.43	57.05	55.59
CIFAR Classification	87.57	82.08	90.74	77.34	70.77
STL RotNet	75.87	40.67	41.11	43.82	42.58
CIFAR RotNet	76.65	54.54	51.11	51.26	54.82
CvD RotNet	70.12	49.39	50.71	50.35	49.00
STL AE	64.66	59.71	55.98	54.62	59.22
CIFAR AE	64.11	59.31	54.75	54.05	59.37
CvD AE	64.47	50.67	51.25	50.71	50.38
STL DAE	64.69	58.19	57.37	55.04	62.04
CIFAR DAE	66.13	58.61	53.92	52.89	59.57
CvD DAE	57.23	50.37	51.64	51.08	51.84

Table C.2: Anomaly detection results for the "dog" class in the Cats vs. Dogs dataset.

		Anomaly Detection Method (AUROC)			
Teacher Representation	**Classification Acc.**	Knowledge Distillation	MSE	Mahalanobis (Diagonal)	Mahalanobis (Full)
Baseline	*83.40*	*90.07*	*89.57*	*76.54*	*86.74*
Random	64.11	53.28	47.99	48.63	48.83
STL Classification	71.28	64.66	65.96	60.56	57.09
CIFAR Classification	87.57	81.88	93.84	79.12	79.05
STL RotNet	75.87	69.15	66.30	60.73	65.34
CIFAR RotNet	76.65	58.83	51.59	51.62	56.87
CvD RotNet	70.12	50.97	47.44	48.23	50.28
STL AE	64.66	44.35	48.33	48.58	44.31
CIFAR AE	64.11	43.94	49.38	48.85	42.83
CvD AE	64.47	51.90	48.61	49.14	50.88
STL DAE	64.69	49.10	45.31	46.73	40.99
CIFAR DAE	66.13	48.23	49.35	49.37	43.43
CvD DAE	57.23	51.37	49.11	49.27	49.48

Table C.3: Anomaly detection results for unimodal CIFAR-10 configuration, averaged over each class.

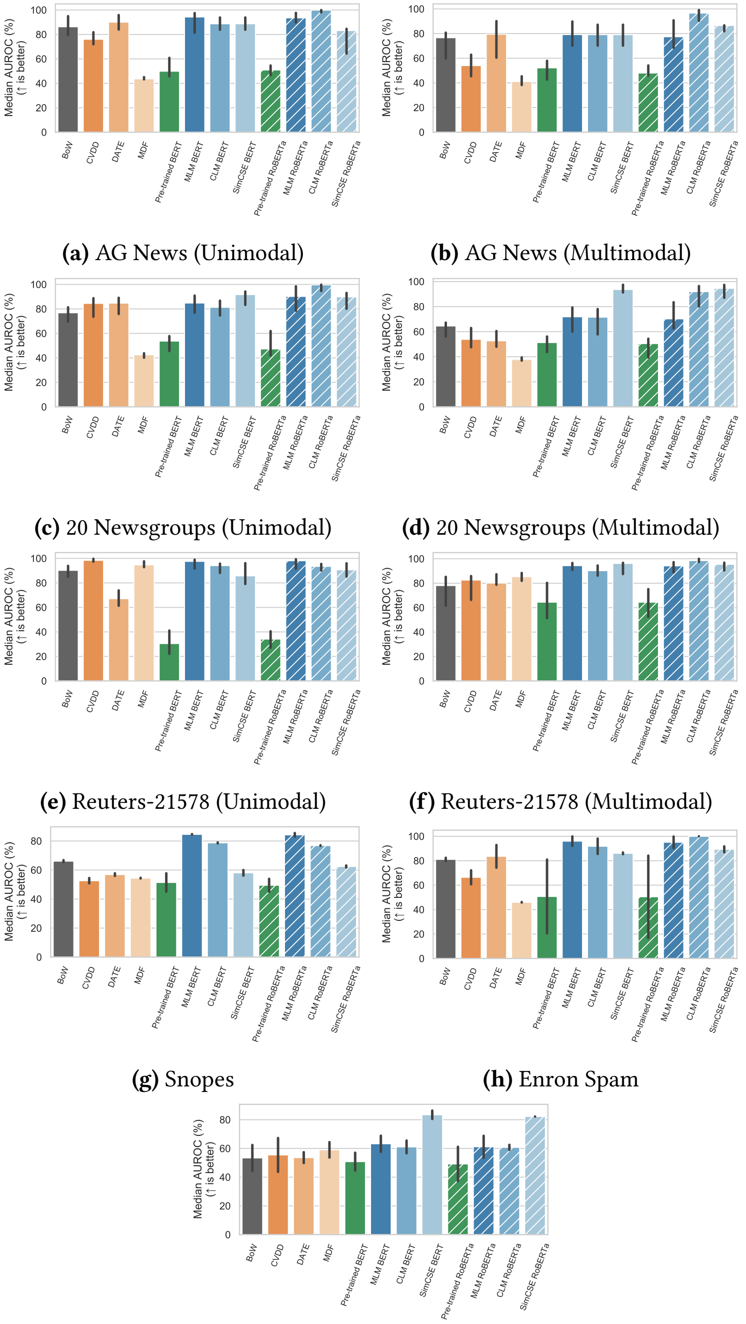
	Anomaly Detection Method (AUROC)			
Teacher Representation	Knowledge Distillation	MSE	Mahalanobis (Diagonal)	Mahalanobis (Full)
Baseline	*95.44*	*94.83*	*94.99*	*89.96*
Random	60.09	58.73	58.69	57.25
STL Classification	85.29	88.45	89.05	82.66
FMNIST Classification	57.60	56.53	56.58	56.60
STL RotNet	70.71	70.28	70.00	66.32
FMNIST RotNet	57.90	54.91	55.32	52.81
CIFAR RotNet	85.35	85.99	85.98	84.88
CIFAR SimCLR	54.18	33.84	34.36	50.85
STL AE	62.85	61.16	62.17	61.03
FMNIST AE	63.50	60.77	59.82	59.24
CIFAR AE	61.03	60.82	61.74	59.33
STL DAE	60.21	62.48	63.00	63.33
FMNIST DAE	60.19	58.98	59.42	57.47
CIFAR DAE	60.06	55.53	57.41	58.52

Table C.4: Anomaly detection results for multimodal CIFAR-10 configuration, averaged over each class.

	Anomaly Detection Method (AUROC)			
Teacher Representation	Knowledge Distillation	MSE	Mahalanobis (Diagonal)	Mahalanobis (Full)
Baseline	*87.60*	*57.20*	*57.39*	*69.57*
Random	50.98	51.16	51.10	51.13
STL Class.	76.28	54.03	55.81	63.33
FMNIST Class.	51.08	50.70	50.77	50.81
STL RotNet	55.31	52.96	52.83	52.90
FMNIST RotNet	51.46	50.37	50.54	50.52
CIFAR RotNet	61.35	59.90	60.16	53.59
CIFAR SimCLR	49.37	31.19	31.31	43.45
STL AE	51.47	51.12	50.77	51.54
FMNIST AE	51.89	51.16	51.46	51.29
CIFAR AE	51.64	51.01	51.55	51.22
STL DAE	51.26	51.36	51.61	52.04
FMNIST DAE	52.31	51.11	51.14	51.06
CIFAR DAE	50.84	50.36	50.04	50.99

C.2 Text

C.2.1 Semantic anomaly detection results



(a) AG News (Unimodal)

(b) AG News (Multimodal)

(c) 20 Newsgroups (Unimodal)

(d) 20 Newsgroups (Multimodal)

(e) Reuters-21578 (Unimodal)

(f) Reuters-21578 (Multimodal)

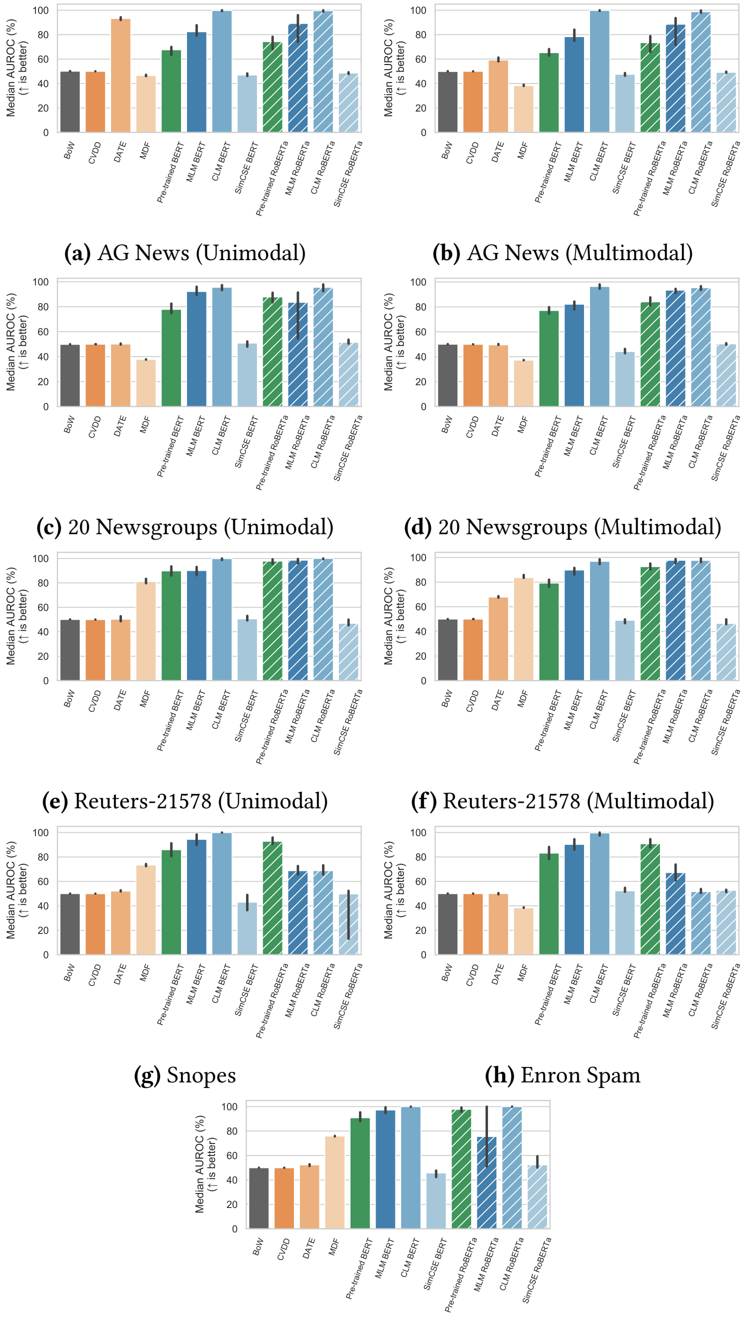
(g) Snopes

(h) Enron Spam

(i) IMDb

Figure C.1: Median semantic anomaly detection results split by dataset.

C.2.2 Word order anomaly detection results



(a) AG News (Unimodal)

(b) AG News (Multimodal)

(c) 20 Newsgroups (Unimodal)

(d) 20 Newsgroups (Multimodal)

(e) Reuters-21578 (Unimodal)

(f) Reuters-21578 (Multimodal)

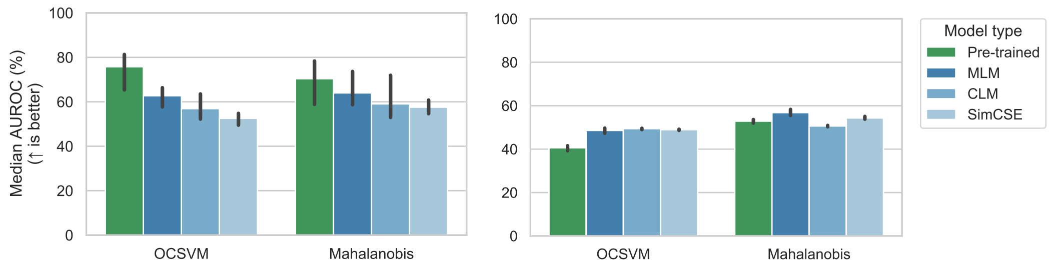
(g) Snopes

(h) Enron Spam

(i) IMDb

Figure C.2: Median word order anomaly detection results split by dataset. The figures include all *n*-gram runs.

C.2.3 Shallow embedding results



(a) Semantic anomaly results. **(b)** Word order anomaly results encompassing all *n*-grams.

Figure C.3: Shallow anomaly detection results using other detectors.

C.3 Speech

C.3.1 Human-level speech deepfake detection results

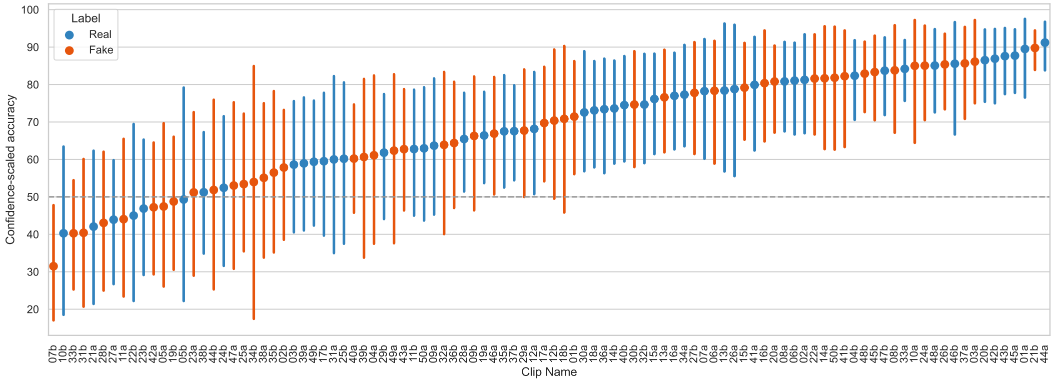


Figure C.4: Confidence-adjusted accuracy scores per clip (English, unary, no familiarisation).

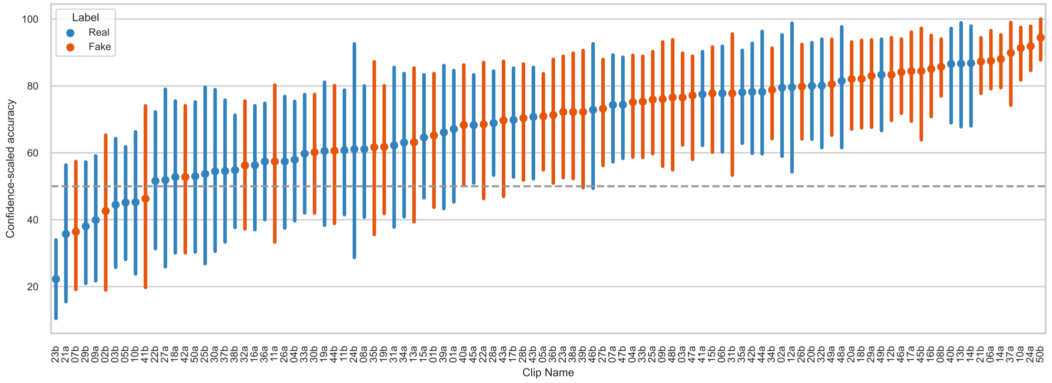


Figure C.5: Confidence-adjusted accuracy scores per clip (English, unary, familiarisation).

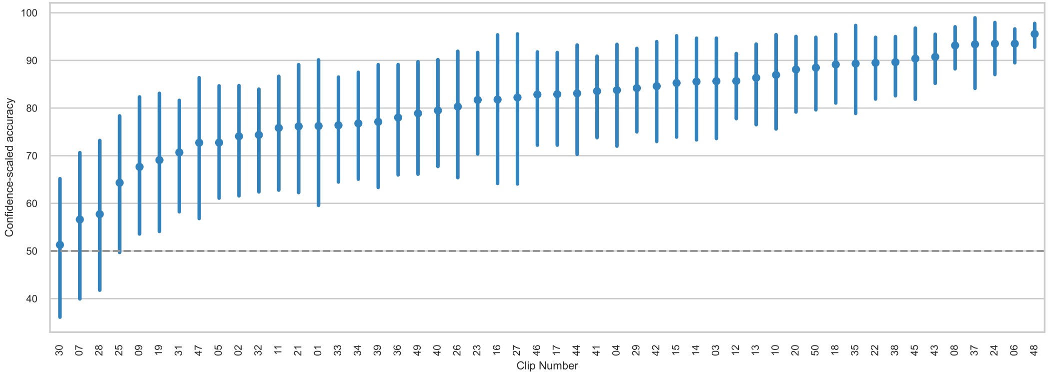


Figure C.6: Confidence-adjusted accuracy scores per clip (English, binary, no familiarisation).

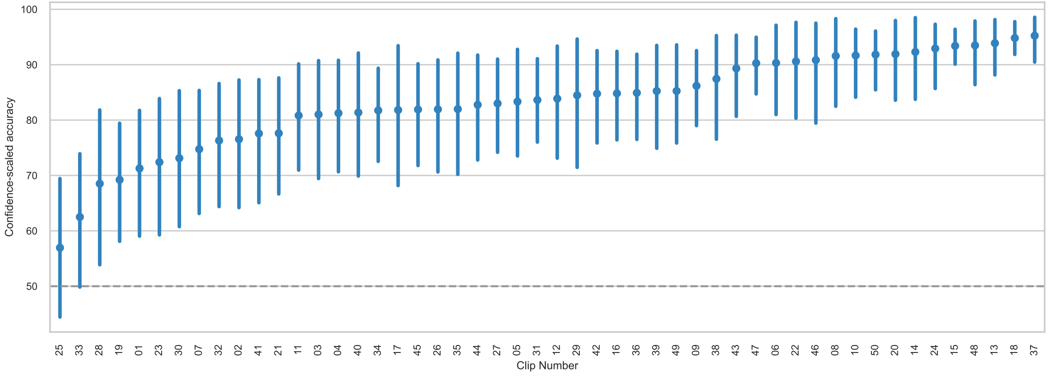


Figure C.7: Confidence-adjusted accuracy scores per clip (English, binary, familiarisation).

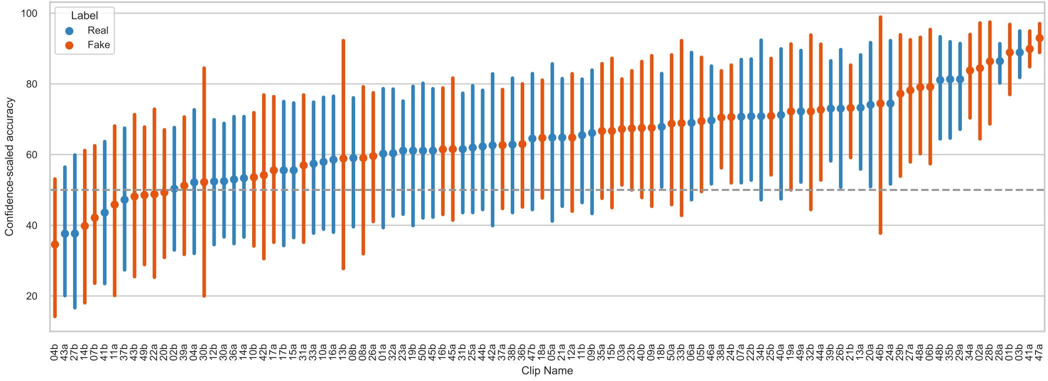


Figure C.8: Confidence-adjusted accuracy scores per clip (Mandarin, unary, no familiarisation).

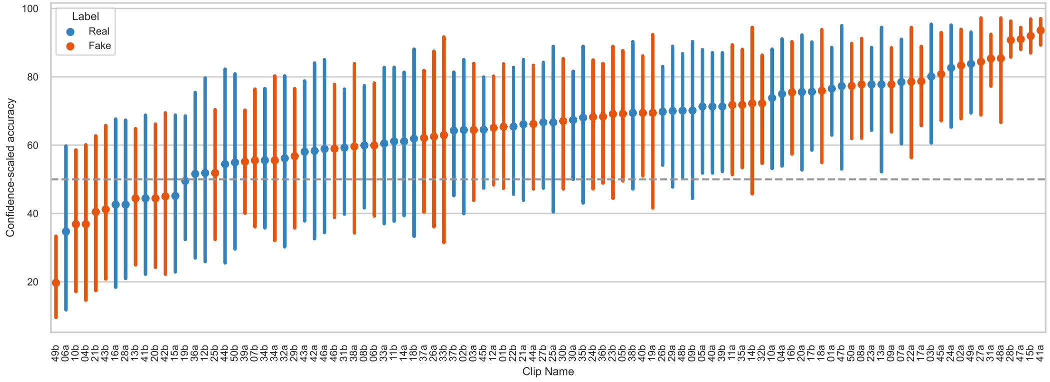


Figure C.9: Confidence-adjusted accuracy scores per clip (Mandarin, unary, familiarisation).

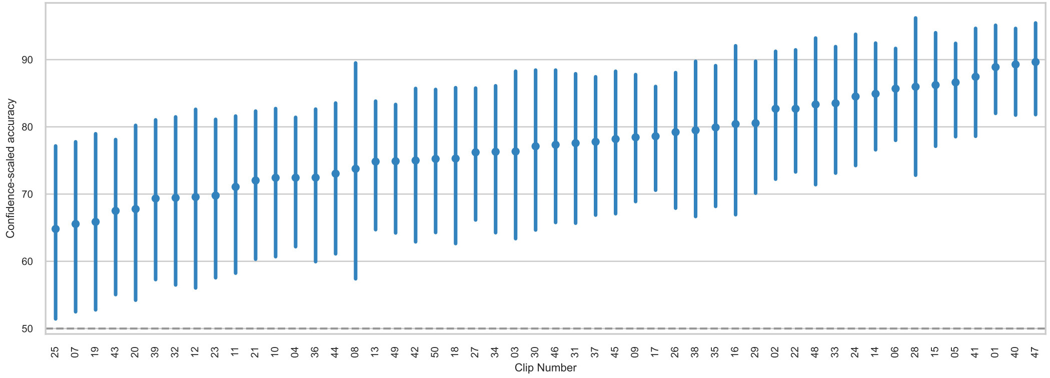


Figure C.10: Confidence-adjusted accuracy scores per clip (Mandarin, binary, no familiarisation).

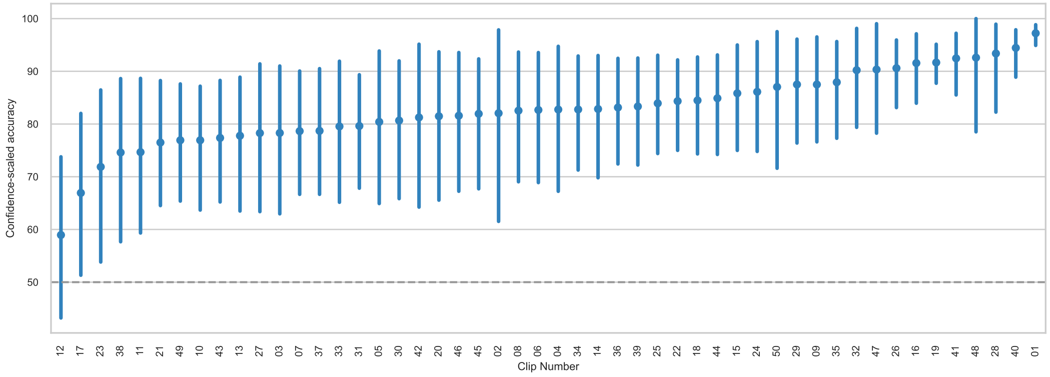


Figure C.11: Confidence-adjusted accuracy scores per clip (Mandarin, binary, familiarisation).

C.3.2 One-class automated speech deepfake detection results

Table C.5: Pre-trained cosine AUROCs (with ranks) for each dataset and representation.

Dataset	**AST**	**SSAST**	**VIT**	**Wav2vec2$_{\mathbf{zh}}$**	**HuBERT$_{\mathbf{zh}}$**	**HuBERT$_{\mathbf{en}}$**	**Wav2vec2$_{\mathbf{en}}$**
ASVspoof 2019 A07	95.20 (22)	96.65 (3)	97.67 (4)	99.34 (19)	99.73 (14)	99.36 (12)	98.27 (6)
ASVspoof 2019 A08	96.80 (19)	89.07 (14)	94.28 (15)	99.48 (17)	99.68 (17)	99.07 (16)	97.16 (9)
ASVspoof 2019 A09	98.22 (14)	83.49 (23)	98.53 (2)	99.60 (15)	99.84 (8)	99.54 (11)	96.87 (13)
ASVspoof 2019 A10	89.03 (31)	94.79 (6)	97.06 (7)	99.21 (20)	99.71 (15)	99.35 (13)	98.15 (7)
ASVspoof 2019 A11	98.72 (11)	92.43 (11)	97.21 (6)	99.63 (13)	99.90 (6)	99.70 (8)	98.93 (5)
ASVspoof 2019 A12	88.22 (32)	87.03 (19)	97.00 (8)	99.12 (22)	99.71 (16)	99.19 (15)	96.65 (15)
ASVspoof 2019 A13	94.72 (23)	97.53 (1)	99.14 (1)	99.62 (14)	99.95 (1)	99.80 (6)	99.07 (3)
ASVspoof 2019 A14	92.34 (27)	83.55 (22)	93.95 (16)	99.48 (18)	99.61 (18)	99.22 (14)	97.26 (8)
ASVspoof 2019 A15	83.37 (38)	82.31 (26)	92.47 (21)	98.67 (23)	98.93 (22)	98.75 (18)	94.75 (20)
ASVspoof 2019 A16	79.29 (44)	88.05 (17)	93.20 (19)	97.29 (25)	98.98 (21)	98.68 (19)	96.31 (17)
ASVspoof 2019 A17	72.24 (49)	58.35 (48)	57.47 (57)	74.11 (53)	74.56 (49)	78.38 (45)	83.18 (38)
ASVspoof 2019 A18	97.77 (16)	54.3 (51)	70.95 (45)	96.75 (28)	96.72 (27)	92.84 (29)	91.76 (27)
ASVspoof 2019 A19	69.65 (54)	61.67 (46)	65.17 (52)	71.59 (54)	63.42 (55)	62.81 (56)	65.44 (55)
ASVspoof 2021 A07	84.68 (37)	81.62 (27)	73.6 (44)	91.83 (35)	90.26 (34)	90.93 (31)	88.82 (31)
ASVspoof 2021 A08	81.78 (42)	73.53 (34)	65.92 (50)	91.59 (36)	86.83 (37)	88.94 (33)	85.13 (33)

Dataset	**AST**	**SSAST**	**VIT**	**Wav2vec2$_{\mathbf{zh}}$**	**HuBERT$_{\mathbf{zh}}$**	**HuBERT$_{\mathbf{en}}$**	**Wav2vec2$_{\mathbf{en}}$**
ASVspoof 2021 A09	87.66 (33)	68.64 (38)	75.2 (42)	94.15 (33)	91.00 (31)	91.27 (30)	84.19 (37)
ASVspoof 2021 A10	76.12 (47)	79.21 (29)	75.53 (41)	90.47 (38)	88.82 (35)	90.27 (32)	87.93 (32)
ASVspoof 2021 A11	91.78 (28)	77.9 (31)	75.94 (40)	97.06 (26)	94.94 (30)	93.14 (28)	90.40 (29)
ASVspoof 2021 A12	70.89 (53)	73.46 (35)	70.14 (46)	87.96 (41)	85.58 (39)	88.77 (34)	84.52 (36)
ASVspoof 2021 A13	81.95 (41)	82.93 (25)	82.71 (32)	96.16 (29)	96.25 (28)	95.67 (26)	91.47 (28)
ASVspoof 2021 A14	68.35 (56)	66.53 (41)	65.93 (49)	91.47 (37)	85.36 (41)	88.16 (36)	82.33 (40)
ASVspoof 2021 A15	63.75 (63)	66.34 (42)	65.56 (51)	86.19 (42)	79.49 (46)	86.87 (39)	78.69 (43)
ASVspoof 2021 A16	65.9 (60)	70.34 (36)	66.45 (48)	82.92 (46)	80.50 (44)	86.75 (40)	84.6 (35)
ASVspoof 2021 A17	66.05 (59)	50.47 (57)	54.74 (60)	63.8 (59)	62.91 (58)	66.89 (53)	73.61 (48)
ASVspoof 2021 A18	86.89 (35)	49.84 (58)	62.83 (53)	77.84 (50)	74.48 (50)	73.91 (47)	75.31 (46)
ASVspoof 2021 A19	65.49 (62)	53.19 (53)	54.62 (61)	63.39 (60)	57.63 (63)	57.19 (62)	61.81 (56)
CFAD AISHELL1 F01	94.07 (25)	86.94 (20)	92.83 (20)	95.27 (31)	95.95 (29)	95.91 (24)	81.29 (41)
CFAD AISHELL1 F02	98.00 (15)	96.38 (5)	93.74 (17)	96.85 (27)	90.96 (32)	87.83 (38)	77.35 (44)
CFAD AISHELL1 F03	89.07 (30)	69.80 (37)	93.58 (18)	79.71 (47)	80.04 (45)	83.73 (43)	72.55 (49)
CFAD AISHELL1 F04	95.78 (21)	88.59 (16)	90.88 (26)	97.38 (24)	97.22 (25)	94.47 (27)	84.74 (34)
CFAD AISHELL1 F05	98.34 (13)	63.23 (43)	91.14 (25)	99.63 (12)	98.93 (23)	98.68 (20)	96.70 (14)
CFAD AISHELL1 F06	96.57 (20)	94.24 (7)	92.23 (23)	93.79 (34)	85.91 (38)	88.51 (35)	72.40 (50)
CFAD AISHELL1 F07	97.45 (17)	50.55 (55)	91.84 (24)	99.20 (21)	97.97 (24)	98.01 (21)	95.15 (19)

Dataset	**AST**	**SSAST**	**VIT**	**Wav2vec2$_{zh}$**	**HuBERT$_{zh}$**	**HuBERT$_{en}$**	**Wav2vec2$_{en}$**
CFAD AISHELL1 F08	99.49 (9)	67.34 (39)	94.83 (14)	99.74 (9)	99.2 (20)	98.94 (17)	96.93 (11)
CFAD AISHELL3 F01	80.47 (43)	62.28 (45)	61.00 (55)	77.12 (51)	66.92 (52)	65.18 (55)	56.97 (58)
CFAD AISHELL3 F02	94.34 (24)	89.26 (13)	86.96 (30)	94.63 (32)	81.66 (42)	68.95 (51)	66.68 (54)
CFAD AISHELL3 F03	86.70 (36)	43.28 (69)	84.6 (31)	78.51 (48)	80.6 (43)	70.44 (49)	60.60 (57)
CFAD AISHELL3 F04	93.38 (26)	76.68 (33)	82.09 (35)	96.04 (30)	96.94 (26)	85.36 (42)	79.29 (42)
CFAD AISHELL3 F05	98.62 (12)	53.99 (52)	90.59 (27)	99.85 (7)	99.82 (11)	97.17 (23)	96.10 (18)
CFAD AISHELL3 F06	82.89 (39)	84.67 (21)	82.18 (34)	77.90 (49)	65.78 (54)	66.24 (54)	55.50 (59)
CFAD AISHELL3 F07	97.26 (18)	47.68 (65)	90.23 (28)	99.59 (16)	99.6 (19)	95.83 (25)	94.44 (21)
CFAD AISHELL3 F08	99.24 (10)	57.90 (49)	92.46 (22)	99.86 (6)	99.83 (10)	97.74 (22)	96.33 (16)
CFAD MagicRead F01	49.13 (68)	77.06 (32)	56.3 (59)	66.99 (57)	56.97 (65)	61.01 (59)	66.92 (53)
CFAD MagicRead F02	72.19 (50)	96.55 (4)	81.29 (36)	83.18 (45)	68.60 (51)	70.64 (48)	75.11 (47)
CFAD MagicRead F03	38.84 (69)	50.48 (56)	57.99 (56)	63.81 (58)	63.32 (56)	62.46 (57)	67.16 (52)
CFAD MagicRead F04	66.72 (58)	88.91 (15)	74.94 (43)	84.77 (44)	78.38 (47)	77.66 (46)	82.89 (39)
CFAD MagicRead F05	74.02 (48)	66.89 (40)	76.71 (39)	90.06 (39)	88.33 (36)	86.05 (41)	92.27 (25)
CFAD MagicRead F06	54.02 (66)	93.73 (9)	80.11 (37)	67.98 (55)	62.92 (57)	68.52 (52)	68.4 (51)
CFAD MagicRead F07	69.44 (55)	60.06 (47)	79.37 (38)	85.17 (43)	85.50 (40)	82.57 (44)	89.92 (30)
CFAD MagicRead F08	87.41 (34)	62.28 (44)	82.4 (33)	88.98 (40)	90.80 (33)	88.09 (37)	93.64 (23)
CFAD THCHS30 F01	99.88 (3)	79.90 (28)	96.59 (10)	99.82 (8)	99.79 (13)	99.76 (7)	93.84 (22)

Dataset	**AST**	**SSAST**	**VIT**	**Wav2vec2$_{\text{zh}}$**	**HuBERT$_{\text{zh}}$**	**HuBERT$_{\text{en}}$**	**Wav2vec2$_{\text{en}}$**
CFAD THCHS30 F02	99.94 (1)	96.84 (2)	98.46 (3)	99.96 (1)	99.93 (4)	99.95 (1)	96.99 (10)
CFAD THCHS30 F03	99.63 (8)	78.34 (30)	88.46 (29)	99.69 (11)	99.81 (12)	99.56 (10)	92.2 (26)
CFAD THCHS30 F04	99.87 (4)	83.06 (24)	96.17 (11)	99.90 (4)	99.89 (7)	99.84 (4)	96.92 (12)
CFAD THCHS30 F05	99.71 (6)	89.69 (12)	95.27 (13)	99.91 (3)	99.94 (3)	99.87 (3)	99.34 (1)
CFAD THCHS30 F06	99.86 (5)	87.51 (18)	97.64 (5)	99.72 (10)	99.83 (9)	99.63 (9)	92.68 (24)
CFAD THCHS30 F07	99.69 (7)	92.56 (10)	95.72 (12)	99.87 (5)	99.93 (5)	99.81 (5)	99.06 (4)
CFAD THCHS30 F08	99.89 (2)	93.99 (8)	96.66 (9)	99.93 (2)	99.95 (2)	99.88 (2)	99.32 (2)
FMFCC	49.86 (67)	49.83 (59)	56.88 (58)	37.55 (69)	36.98 (69)	33.91 (69)	24.58 (69)
WaveFake JSUT Multi Band MelGAN	62.32 (64)	47.28 (66)	52.69 (63)	55.28 (66)	56.19 (66)	56.62 (63)	53.18 (62)
WaveFake JSUT Parallel WaveGAN	55.66 (65)	50.64 (54)	48.01 (67)	58.59 (62)	57.83 (62)	56.4 (64)	50.68 (68)
WaveFake LJSpeech Conformer	91.33 (29)	48.12 (63)	62.39 (54)	76.83 (52)	76.86 (48)	70.04 (50)	75.38 (45)
WaveFake LJSpeech Full Band MelGAN	70.98 (52)	48.11 (64)	50.14 (65)	52.43 (67)	52.79 (67)	55.98 (66)	51.07 (67)
WaveFake LJSpeech HiFiGAN	67.95 (57)	48.17 (62)	44.82 (69)	51.49 (68)	52.34 (68)	54.21 (68)	51.27 (66)
WaveFake LJSpeech MelGAN	65.49 (61)	45.81 (68)	52.82 (62)	58.66 (61)	58.11 (61)	55.21 (67)	51.31 (65)
WaveFake LJSpeech MelGAN Large	82.77 (40)	45.86 (67)	68.57 (47)	58.35 (64)	59.65 (60)	56.12 (65)	51.73 (64)
WaveFake LJSpeech Multi Band MelGAN	77.16 (46)	48.89 (60)	50.91 (64)	58.56 (63)	60.75 (59)	61.79 (58)	53.34 (61)
WaveFake LJSpeech Parallel WaveGAN	71.11 (51)	56.18 (50)	46.72 (68)	57.99 (65)	57.22 (64)	58.51 (61)	53.14 (63)

Dataset	AST	SSAST	VIT	Wav2vec2$_{\text{zh}}$	HuBERT$_{\text{zh}}$	HuBERT$_{\text{en}}$	Wav2vec2$_{\text{en}}$
WaveFake LJSpeech Waveglow	77.81 (45)	48.62 (61)	48.12 (66)	67.10 (56)	65.88 (53)	59.07 (60)	53.73 (60)

Table C.6: Shallow feature-engineered cosine AUROCs (with ranks) for each dataset

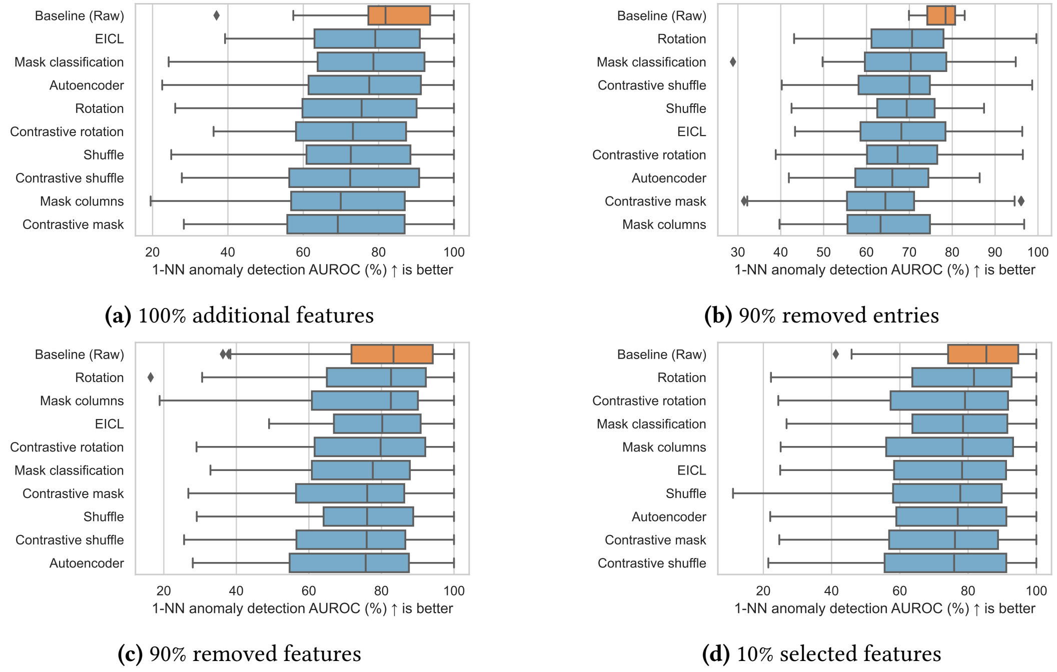
Dataset	Raw waveform	LFCC	Mel spectrogram	MFCC	STFT
ASVspoof 2019 A07	81.17 (18)	99.9 (2)	99.96 (1)	99.97 (2)	92.11 (5)
ASVspoof 2019 A08	78.57 (25)	99.17 (9)	99.48 (10)	99.29 (10)	83.62 (25)
ASVspoof 2019 A09	86.85 (7)	99.7 (8)	99.96 (2)	99.88 (7)	91.93 (6)
ASVspoof 2019 A10	82.10 (15)	99.87 (3)	99.93 (7)	99.95 (3)	90.6 (10)
ASVspoof 2019 A11	84.92 (11)	99.72 (7)	99.95 (4)	99.92 (4)	91.46 (9)
ASVspoof 2019 A12	81.55 (16)	99.74 (6)	99.94 (6)	99.87 (8)	89.9 (11)
ASVspoof 2019 A13	90.28 (3)	99.97 (1)	99.93 (8)	99.99 (1)	94.01 (2)
ASVspoof 2019 A14	89.9 (4)	99.75 (5)	99.95 (3)	99.91 (5)	91.83 (7)
ASVspoof 2019 A15	85.86 (8)	99.76 (4)	99.95 (5)	99.89 (6)	92.91 (3)
ASVspoof 2019 A16	81.13 (20)	99.16 (10)	99.84 (9)	99.56 (9)	85.36 (21)
ASVspoof 2019 A17	52.08 (55)	53.51 (53)	57.26 (49)	54.36 (51)	48.07 (64)
ASVspoof 2019 A18	61.14 (41)	57.37 (45)	60.21 (46)	56.37 (45)	49.2 (61)
ASVspoof 2019 A19	59.24 (44)	54.99 (49)	56.01 (51)	55.45 (47)	57.49 (41)
ASVspoof 2021 A07	77.17 (29)	93.74 (18)	94.23 (17)	93.18 (18)	86.5 (17)
ASVspoof 2021 A08	75.66 (31)	91.22 (28)	91.35 (29)	89.84 (30)	78.61 (30)
ASVspoof 2021 A09	81.29 (17)	92.21 (23)	93.17 (21)	90.8 (26)	86.49 (18)

Dataset	Raw waveform	LFCC	Mel spectrogram	MFCC	STFT
ASVspoof 2021 A10	78.58 (24)	93.44 (20)	93.96 (19)	92.86 (19)	85.15 (23)
ASVspoof 2021 A11	81.17 (19)	92.79 (21)	93.84 (20)	92.21 (20)	85.47 (20)
ASVspoof 2021 A12	78.58 (23)	92.22 (22)	92.71 (25)	90.92 (23)	85.23 (22)
ASVspoof 2021 A13	86.86 (6)	94.72 (16)	94.57 (16)	95.02 (15)	89.76 (12)
ASVspoof 2021 A14	87.04 (5)	92.17 (25)	92.89 (23)	90.89 (24)	86.78 (16)
ASVspoof 2021 A15	83.47 (13)	92.19 (24)	92.85 (24)	90.85 (25)	88.42 (13)
ASVspoof 2021 A16	77.48 (28)	90.43 (31)	91.46 (28)	89.88 (28)	80.01 (28)
ASVspoof 2021 A17	52.7 (53)	50.61 (56)	52.47 (53)	51.25 (52)	51.61 (47)
ASVspoof 2021 A18	59.71 (43)	54.88 (51)	56.08 (50)	54.81 (49)	51.36 (49)
ASVspoof 2021 A19	57.26 (46)	54.66 (52)	55.07 (52)	55.39 (48)	56.09 (42)
CFAD Aishell1 F01	75.45 (32)	92.14 (26)	90.01 (30)	91.27 (22)	71.17 (36)
CFAD Aishell1 F02	73.71 (33)	97.69 (12)	97.37 (11)	97.47 (12)	68.71 (38)
CFAD Aishell1 F03	70.67 (34)	72.51 (37)	66.39 (42)	68.84 (37)	67.24 (39)
CFAD Aishell1 F04	85.14 (9)	95.23 (15)	95.07 (15)	94.58 (16)	82.91 (26)
CFAD Aishell1 F05	90.66 (2)	77.35 (35)	83.54 (34)	74.19 (35)	97.42 (1)
CFAD Aishell1 F06	77.78 (27)	97.99 (11)	97.32 (12)	97.68 (11)	73.16 (35)
CFAD Aishell1 F07	84.17 (12)	70.34 (39)	72.00 (38)	68.14 (39)	77.01 (32)
CFAD Aishell1 F08	93.53 (1)	56.03 (48)	62.80 (44)	54.44 (50)	91.54 (8)

Dataset	Raw waveform	LFCC	Mel spectrogram	MFCC	STFT
CFAD Aishell3 F01	52.98 (52)	70.81 (38)	70.63 (39)	70.30 (36)	52.16 (45)
CFAD Aishell3 F02	52.36 (54)	93.75 (17)	92.09 (26)	93.47 (17)	52.67 (44)
CFAD Aishell3 F03	51.10 (59)	50.69 (55)	52.33 (54)	49.86 (54)	51.13 (51)
CFAD Aishell3 F04	68.45 (36)	85.72 (33)	88.28 (31)	84.49 (32)	69.53 (37)
CFAD Aishell3 F05	80.6 (21)	66.55 (42)	78.43 (36)	67.57 (40)	92.69 (4)
CFAD Aishell3 F06	51.82 (57)	90.85 (29)	86.47 (32)	91.32 (21)	51.48 (48)
CFAD Aishell3 F07	51.84 (56)	44.34 (67)	57.79 (48)	47.65 (57)	49.97 (55)
CFAD Aishell3 F08	82.33 (14)	56.45 (46)	65.62 (43)	56.19 (46)	83.83 (24)
CFAD MagicRead F01	54.01 (48)	87.65 (32)	84.39 (33)	84.24 (33)	50.26 (54)
CFAD MagicRead F02	53.84 (49)	97.51 (13)	96.83 (13)	95.73 (13)	49.53 (59)
CFAD MagicRead F03	51.82 (58)	69.99 (41)	62.46 (45)	66.48 (42)	46.58 (67)
CFAD MagicRead F04	68.75 (35)	93.53 (19)	91.63 (27)	90.05 (27)	67.15 (40)
CFAD MagicRead F05	84.96 (10)	70.22 (40)	67.67 (41)	63.34 (43)	41.91 (69)
CFAD MagicRead F06	53.72 (50)	97.25 (14)	96.12 (14)	95.48 (14)	50.3 (53)
CFAD MagicRead F07	57.78 (45)	65.56 (43)	58.42 (47)	57.1 (44)	52.83 (43)
CFAD MagicRead F08	61.07 (42)	54.89 (50)	52.23 (55)	49.92 (53)	74.9 (34)
CFAD THCHS30 F01	65.10 (39)	74.43 (36)	76.94 (37)	68.27 (38)	77.44 (31)
CFAD THCHS30 F02	67.64 (38)	91.70 (27)	92.94 (22)	89.85 (29)	78.82 (29)

Dataset	Raw waveform	LFCC	Mel spectrogram	MFCC	STFT
CFAD THCHS30 F03	64.76 (40)	50.81 (54)	47.57 (57)	37.03 (65)	75.46 (33)
CFAD THCHS30 F04	78.18 (26)	82.03 (34)	80.51 (35)	76.22 (34)	86.18 (19)
CFAD THCHS30 F05	78.71 (22)	56.13 (47)	44.05 (59)	36.92 (66)	46.49 (68)
CFAD THCHS30 F06	68.12 (37)	90.60 (30)	94.00 (18)	88.71 (31)	80.58 (27)
CFAD THCHS30 F07	75.82 (30)	47.65 (60)	29.93 (69)	26.25 (68)	86.81 (15)
CFAD THCHS30 F08	46.03 (67)	45.86 (64)	32.86 (68)	25.73 (69)	88.07 (14)
FMFCC	43.23 (68)	64.11 (44)	68.2 (40)	66.58 (41)	47.87 (65)
WaveFake JSUT Multi Band MelGAN	54.11 (47)	50.01 (57)	45.84 (58)	47.78 (56)	51.98 (46)
WaveFake JSUT Parallel WaveGAN	53.31 (51)	47.49 (62)	48.84 (56)	47.90 (55)	51.21 (50)
WaveFake LJSpeech Conformer	38.66 (69)	43.51 (68)	41.34 (63)	47.46 (58)	46.80 (66)
WaveFake LJSpeech Full Band MelGAN	48.34 (66)	45.58 (65)	41.46 (62)	44.78 (62)	48.81 (62)
WaveFake LJSpeech HiFiGAN	50.93 (60)	47.55 (61)	40.25 (64)	45.21 (61)	50.44 (52)
WaveFake LJSpeech MelGAN	49.33 (62)	47.45 (63)	41.91 (60)	45.37 (60)	49.71 (57)
WaveFake LJSpeech MelGAN Large	48.65 (65)	47.66 (59)	41.55 (61)	45.48 (59)	49.58 (58)
WaveFake LJSpeech Multi Band MelGAN	50.72 (61)	48.09 (58)	38.5 (65)	44.32 (63)	49.51 (60)
WaveFake LJSpeech Parallel WaveGAN	49.13 (63)	38.59 (69)	34.44 (67)	36.08 (67)	49.91 (56)
WaveFake LJSpeech Waveglow	48.67 (64)	44.39 (66)	36.11 (66)	42.48 (64)	48.37 (63)

C.4 Tabular data



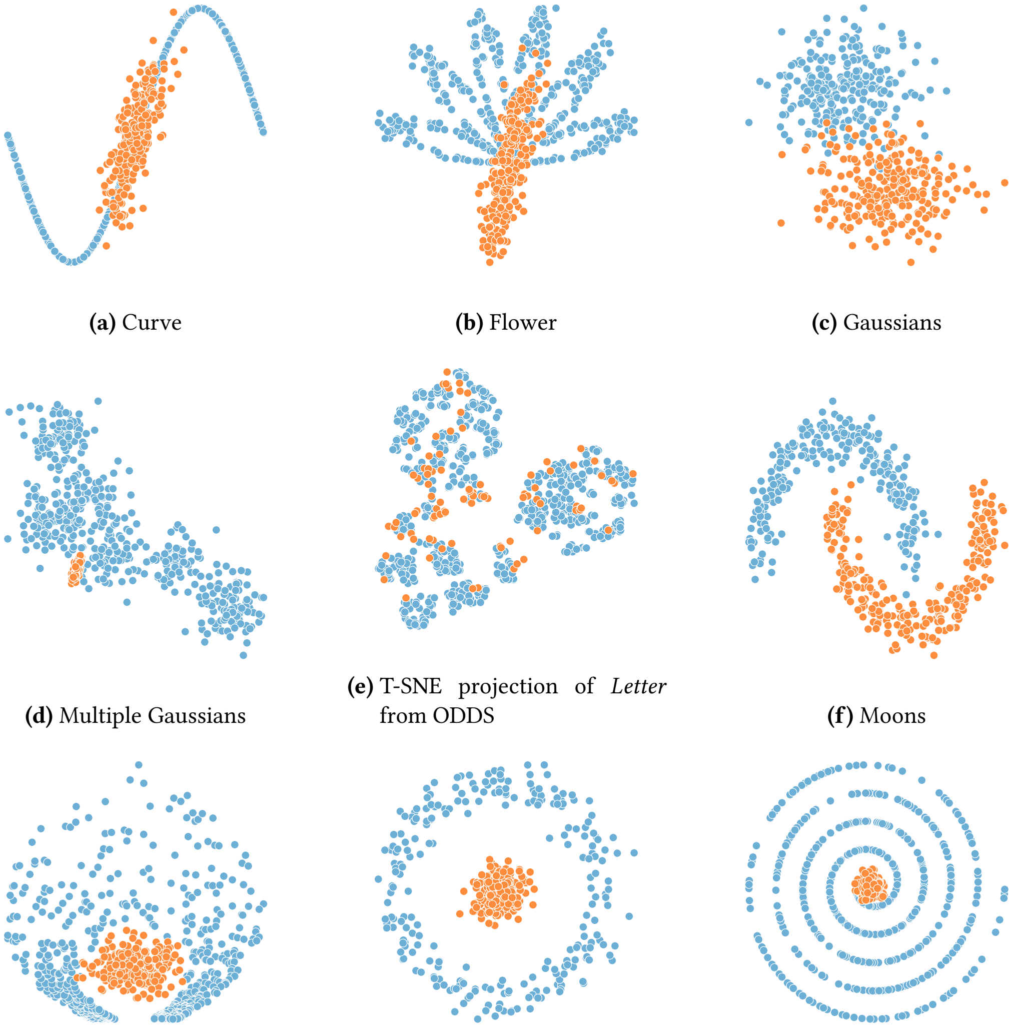
(a) 100% additional features

(b) 90% removed entries

(c) 90% removed features

(d) 10% selected features

Figure C.12: Box plot comparing nearest neighbour AUROCs for each of the self-supervised pretext tasks on corrupted input data.



(a) Curve

(b) Flower

(c) Gaussians

(d) Multiple Gaussians

(e) T-SNE projection of *Letter* from ODDS

(f) Moons

(g) Ring

(h) Pinched ring

(i) Spiral

Figure C.13: Illustrations of the toy test data. Blue points are normal whereas orange points are anomalous.

Bibliography

Bibliography

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

[2] Neslihan Kose and Jean-Luc Dugelay. On the vulnerability of face recognition systems to spoofing mask attacks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2357–2361, 2013.

[3] Federal Bureau of Investigation. Malicious actors manipulating photos and videos to create explicit content and sextortion schemes, June 2023. Public Service Announcement (I-060523-PSA); `https://www.ic3.gov/Media/Y2023/PSA230605`; accessed September 27 2023.

[4] Federal Bureau of Investigation. Deepfakes and stolen pii utilized to apply for remote work positions, June 2022. Public Service Announcement (I-062822-PSA); `https://www.ic3.gov/Media/Y2022/PSA220628`; accessed September 27 2023.

[5] Thomas Brewster. Fraudsters cloned company director's voice in $35 million bank heist, police find, November 2022. `https://www.forbes.com/sites/thomasbrewster/2021/10/14/huge-bank-fraud-uses-deep-fake-voice-tech-to-steal-millions/?sh=7dfbccf67559`; accessed January 19 2023.

[6] Catherine Stupp. Fraudsters used ai to mimic ceo's voice in unusual cybercrime case, August 2019. `https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402`; accessed January 19 2023.

[7] Europol. *Law enforcement and the challenge of deepfakes.* April 2022. Accessed September 27 2023.

[8] Kimberly T. Mai, Sergi Bray, Toby Davies, and Lewis D. Griffin. Warning: Humans cannot reliably detect speech deepfakes. *PLoS One*, 18(8):e0285333, 2023.

[9] Nils C Köbis, Barbora Doležalová, and Ivan Soraperra. Fooled twice: People cannot detect deepfakes but think they can. *Iscience*, 24(11), 2021.

[10] Juniper Lovato, Jonathan St-Onge, Randall Harp, Gabriela Salazar Lopez, Sean P. Rogers, Ijaz Ul Haq, Laurent Hébert-Dufresne, and Jeremiah Onaolapo. Diverse misinformation: impacts of human biases on detection of deepfakes on networks. *npj Complexity*, 1(1):5, 2024.

[11] Daniela Buser, Adrian Schwaninger, Juergen Sauer, and Yanik Sterchi. Time on task and task load in visual inspection: A four-month field study with x-ray baggage screeners. *Applied Ergonomics*, 111:103995, 2023.

[12] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436. IEEE Computer Society, 2015.

[13] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

[14] Nicolas M. Müller, Pavel Czempin, Franziska Dieckmann, Adam Froghyar, and Konstantin Böttinger. Does audio deepfake detection generalize? In *Annual Conference of the International Speech Communication Association*, pages 2783–2787, 2022.

[15] Tal Reiss, Niv Cohen, Eliahu Horwitz, Ron Abutbul, and Yedid Hoshen. Anomaly detection requires better representations. In *European Conference on Computer Vision Workshops*, volume 13804 of *Lecture Notes in Computer Science*, pages 56–68. Springer, 2022.

[16] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20827–20840. PMLR, 2022.

[17] Lewis D. Griffin, Matthew Caldwell, Jerone T. A. Andrews, and Helene Bohler. "unexpected item in the bagging area": Anomaly detection in x-ray security images. *IEEE Transactions on Information Forensics and Security*, 14(6):1539–1553, 2019.

[18] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54(2):38:1–38:38, 2022.

[19] Larry Greenmeier. Exposing the weakest link: As airline passenger security tightens, bombers target cargo holds. *Scientific American*, November 2010. `https://www.scientificamerican.com/article/aircraft-cargo-bomb-security/`; accessed October 31 2023.

[20] Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R. Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, A. Taylan Cemgil, S. M. Ali Eslami, and Olaf Ronneberger. Contrastive training for improved out-of-distribution detection. *CoRR*, abs/2007.05566, 2020.

[21] Alex Krizhevsky and Geoffrey E. Hinton. Learning multiple layers of features from tiny images. *Technical report*, 2009.

[22] Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. A unified survey on anomaly, novelty, open-set, and out of-distribution detection: Solutions and future challenges. *Transactions on Machine Learning Research*, 2022.

[23] Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. In *International Conference on Learning Representations*, 2020.

[24] Md. Sahidullah, Tomi Kinnunen, and Cemal Hanilçi. A comparison of features for synthetic speech detection. In *Annual Conference of the International Speech Communication Association*, pages 2087–2091. ISCA, 2015.

[25] Oscar Déniz, Gloria Bueno García, Jesús Salido, and Fernando De la Torre. Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603, 2011.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification

with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012.

[27] Karl R. Weiss, Taghi M. Khoshgoftaar, and Dingding Wang. A survey of transfer learning. *Journal of Big Data*, 3:9, 2016.

[28] Kamalesh Palanisamy, Dipika Singhania, and Angela Yao. Rethinking CNN models for audio classification. *CoRR*, abs/2007.11154, 2020.

[29] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Grégoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning. *CoRR*, abs/2304.12210, 2023.

[30] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, volume 9907 of *Lecture Notes in Computer Science*, pages 649–666. Springer, 2016.

[31] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2227–2237. Association for Computational Linguistics, 2018.

[32] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *Technical report*, 2018.

[33] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

[34] Katherine L. Hermann, Ting Chen, and Simon Kornblith. The origins and prevalence of texture bias in convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2020.

[35] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representa-

tion learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[36] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, volume 9910 of *Lecture Notes in Computer Science*, pages 69–84. Springer, 2016.

[37] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544. IEEE Computer Society, 2016.

[38] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15979–15988. IEEE, 2022.

[39] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[40] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

[41] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

[42] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM, 2015.

[43] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[44] Ricards Marcinkevics and Julia E. Vogt. Interpretable and explainable machine learn-

ing: A methods-centric overview with concrete examples. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(3), 2023.

[45] Jerone TA Andrews, Edward J Morton, and Lewis D Griffin. Detecting anomalous data using auto-encoders. *International Journal of Machine Learning and Computing*, 6(1):21, 2016.

[46] Ki Hyun Kim, Sangwoo Shim, Yongsub Lim, Jongseob Jeon, Jeongwoo Choi, Byungchan Kim, and Andre S. Yoon. Rapp: Novelty detection with reconstruction along projection pathway. In *International Conference on Learning Representations0*, 2020.

[47] Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Advances in Neural Information Processing Systems*, 2021.

[48] Julian Bitterwolf, Maximilian Müller, and Matthias Hein. In or out? fixing imagenet out-of-distribution detection evaluation. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2471–2506. PMLR, 2023.

[49] Antonio Torralba, Robert Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[50] Abeba Birhane and Vinay Uday Prabhu. Large image datasets: A pyrrhic win for computer vision? In *IEEE Winter Conference on Applications of Computer Vision*, pages 1536–1546. IEEE, 2021.

[51] Laurens Van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[52] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019.

[53] Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and GPT-2 embeddings. In *Conference on Empir-*

ical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing, pages 55–65. Association for Computational Linguistics, 2019.

[54] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Conference on Empirical Methods in Natural Language Processing*, pages 9119–9130. Association for Computational Linguistics, 2020.

[55] Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*, 2020.

[56] Olga Kovaleva, Saurabh Kulshreshtha, Anna Rogers, and Anna Rumshisky. BERT busters: Outlier dimensions that disrupt transformers. In *Findings of the Association for Computational Linguistics*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3392–3405. Association for Computational Linguistics, 2021.

[57] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Matteo Carandini, and Kenneth D. Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765):361–365, 2019.

[58] Daniel L. Ruderman. The statistics of natural images. *Network: computation in neural systems*, 5(4):517, 1994.

[59] Kumar Krishna Agrawal, Arnab Kumar Mondal, Arna Ghosh, and Blake A. Richards. α-req : Assessing representation quality in self-supervised learning by measuring eigenspectrum decay. In *Advances in Neural Information Processing Systems*, 2022.

[60] Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10929–10974. PMLR, 2023.

[61] Alexander C. Li, Alexei A. Efros, and Deepak Pathak. Understanding collapse in non-contrastive siamese representation learning. In *European Conference in Computer*

Vision, volume 13691 of *Lecture Notes in Computer Science*, pages 490–505. Springer, 2022.

[62] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.

[63] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 15750–15758. IEEE, 2021.

[64] Carl-Johann Simon-Gabriel, Yann Ollivier, Léon Bottou, Bernhard Schölkopf, and David Lopez-Paz. First-order adversarial vulnerability of neural networks and input dimension. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5809–5817. PMLR, 2019.

[65] David M. J. Tax and Robert P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

[66] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert A. Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4390–4399. PMLR, 2018.

[67] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. PANDA: adapting pretrained features for anomaly detection and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2806–2814. IEEE, 2021.

[68] Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. How to exploit hyperspherical embeddings for out-of-distribution detection? In *International Conference on Learning Representations*, 2023.

[69] Arthur Gretton, Olivier Bousquet, Alexander J. Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic Learning Theory*, volume 3734 of *Lecture Notes in Computer Science*, pages 63–77. Springer, 2005.

[70] Tianyu Cui, Yogesh Kumar, Pekka Marttinen, and Samuel Kaski. Deconfounded representation similarity for comparison of neural networks. In *Advances in Neural Information Processing Systems*, 2022.

[71] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Workshop on Machine Learning for Sensory Data Analysis*, page 4. ACM, 2014.

[72] Samet Akcay, Amir Atapour Abarghouei, and Toby P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian Conference on Computer Vision*, volume 11363 of *Lecture Notes in Computer Science*, pages 622–637. Springer, 2018.

[73] Eric T. Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Görür, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019.

[74] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, pages 14680–14691, 2019.

[75] Ziyu Wang, Bin Dai, David Wipf, and Jun Zhu. Further analysis of outlier detection with deep generative models. In *"I Can't Believe It's Not Better!" at Advances in Neural Information Processing Systems Workshops*, volume 137 of *Proceedings of Machine Learning Research*, pages 11–20. PMLR, 2020.

[76] Robin Schirrmeister, Yuxuan Zhou, Tonio Ball, and Dan Zhang. Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. In *Advances in Neural Information Processing Systems*, 2020.

[77] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *CoRR*, abs/1802.06222, 2018.

[78] Xiaoyi Gu, Leman Akoglu, and Alessandro Rinaldo. Statistical analysis of nearest neighbor methods for anomaly detection. In *Advances in Neural Information Processing Systems*, pages 10921–10931, 2019.

[79] Liron Bergman, Niv Cohen, and Yedid Hoshen. Deep nearest neighbor anomaly detection. *CoRR*, abs/2002.10445, 2020.

[80] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, pages 9781–9791, 2018.

[81] Keyang Xu, Tongzheng Ren, Shikun Zhang, Yihao Feng, and Caiming Xiong. Unsupervised out-of-domain detection via pre-trained transformers. In *Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 1052–1061. Association for Computational Linguistics, August 2021.

[82] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186. Association for Computational Linguistics, 2019.

[83] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[84] Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, Wenxuan Peng, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. Openood: Benchmarking generalized out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 2022.

[85] Daniel Pérez-Cabo, David Jimenez-Cabello, Artur Costa-Pazo, and Roberto Javier López-Sastre. Deep anomaly detection for generalized face anti-spoofing. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1591–1600. IEEE, 2019.

[86] Selim S Sarikan and A Murat Ozbayoglu. Anomaly detection in vehicle traffic with image processing and machine learning. *Procedia Computer Science*, 140:64–69, 2018.

[87] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[88] Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

[89] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Conference on Computer Vision and Pattern Recognition*, pages 4182–4191, 2020.

[90] Kamil Ciosek, Vincent Fortuin, Ryota Tomioka, Katja Hofmann, and Richard E. Turner. Conservative uncertainty estimation by fitting prior networks. In *International Conference on Learning Representations*, 2020.

[91] Kimberly T. Mai, Toby Davies, and Lewis D. Griffin. Brittle features may help anomaly detection. *Women in Computer Vision Workshop at Conference on Computer Vision and Pattern Recognition*, 2021.

[92] Seong Soo Kim and A. L. Narasimha Reddy. Image-based anomaly detection technique: Algorithm, implementation and effectiveness. *IEEE Journal on Selected Areas in Communications*, 24(10):1942–1954, 2006.

[93] Erik Rodner, Esther-Sabrina Wacker, Michael Kemmler, and Joachim Denzler. One-class classification for anomaly detection in wire ropes with gaussian processes in a few lines of code. In *IAPR Conference on Machine Vision Applications*, pages 219–222, 2011.

[94] Jerone Andrews, Thomas Tanay, Edward J Morton, and Lewis D Griffin. Transfer representation-learning for anomaly detection. In *International Conference on Machine Learning*, 2016.

[95] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE Computer Society, 2009.

[96] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018.

[97] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations*, 2015.

[98] Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? In *International Conference on Learning Representations*, 2019.

[99] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[100] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

[101] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016.

[102] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

[103] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and D. Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org, 2010.

[104] Jeremy Elson, John Douceur, Jon Howell, and Jared Saul. Asirra: A captcha that exploits interest-aligned manual image categorization. In *ACM Conference on Computer and Communications Security*. Association for Computing Machinery, Inc., October 2007.

[105] Christine Kaeser-Chen, Fruit Pathology, and Maggie Sohier Dane. Plant pathology, 2020. `https://kaggle.com/competitions/plant-pathology-2020-fgvc7`.

[106] Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík, editors, *International Conference on Artificial Intelligence and Statistics*, volume 15 of *JMLR Proceedings*, pages 215–223, 2011.

[107] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[108] David P. Hughes and Marcel Salathé . An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. *CoRR*, abs/1511.08060, 2015.

[109] George A. Miller. WordNet: A lexical database for English. In *Communications of the ACM*, 1994.

[110] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2015.

[111] David C. Page. Cifar10-fast. `https://github.com/davidcpage/cifar10-fast`, 2018.

[112] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Conference on Computer Vision and Pattern Recognition*, pages 2261–2269. IEEE Computer Society, 2017.

[113] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, pages 15637–15648, 2019.

[114] Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. Towards few-shot fact-checking via perplexity. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1971–1981. Association for Computational Linguistics, June 2021.

[115] Michael Crawford, Taghi M. Khoshgoftaar, Joseph D. Prusa, Aaron N. Richter, and Hamzah Al Najada. Survey of review spam detection using machine learning techniques. *Journal of Big Data*, 2(1):1–24, 2015.

[116] Lukas Ruff, Yury Zemlyanskiy, Robert A. Vandermeulen, Thomas Schnake, and Marius Kloft. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In *Conference of the Association for Computational Linguistics*, pages 4061–4071. Association for Computational Linguistics, 2019.

[117] Udit Arora, William Huang, and He He. Types of out-of-distribution texts and how to detect them. In *Conference on Empirical Methods in Natural Language Processing*, pages 10687–10701. Association for Computational Linguistics, November 2021.

[118] Xiaoya Li, Jiwei Li, Xiaofei Sun, Chun Fan, Tianwei Zhang, Fei Wu, Yuxian Meng, and Jun Zhang. *k*Folden: *k*-fold ensemble for out-of-distribution detection. In *Conference on Empirical Methods in Natural Language Processing*, pages 3102–3115. Association for Computational Linguistics, November 2021.

[119] Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. Revisiting mahalanobis distance for transformer-based out-of-domain detection. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 13675–13682, 2021.

[120] Wenxuan Zhou, Fangyu Liu, and Muhao Chen. Contrastive out-of-distribution detection for pretrained transformers. In *Conference on Empirical Methods in Natural Language Processing*, pages 1100–1111. Association for Computational Linguistics, November 2021.

[121] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[122] Larry M. Manevitz and Malik Yousef. One-class svms for document classification. *Journal of Machine Learning Research*, 2:139–154, 2001.

[123] Daniel Jurafsky and James H. Martin. *Speech and language processing - an introduction to natural language processing, computational linguistics, and speech recognition.* Prentice Hall series in artificial intelligence. Prentice Hall, 2000.

[124] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation

of word representations in vector space. In *International Conference on Learning Representations*, 2013.

[125] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543. ACL, 2014.

[126] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[127] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[128] Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness. In *Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751. Association for Computational Linguistics, July 2020.

[129] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *Technical report*, 1(8):9, 2019.

[130] Andrei Manolache, Florin Brad, and Elena Burceanu. DATE: Detecting anomalies in text via self-supervision of transformers. In *Conference of the North American Chapter of the Association for Computational Linguistics:*, pages 267–277. Association for Computational Linguistics, June 2021.

[131] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020.

[132] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack

Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

[133] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in Neural Information Processing Systems*, 33:11839–11852, 2020.

[134] Vikash Sehwag, Mung Chiang, and Prateek Mittal. SSD: A unified framework for self-supervised outlier detection. In *International Conference on Learning Representations*, 2021.

[135] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910. Association for Computational Linguistics, November 2021.

[136] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *International Conference on Computer Vision*, pages 19–27. IEEE Computer Society, 2015.

[137] Ken Lang. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart J. Russell, editors, *International Conference on Machine Learning*, pages 331–339, 1995.

[138] David D. Lewis. Reuters-21578 text categorization test collection, distribution 1.0, 1997.

[139] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[140] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics*, pages 142–150. Association for Computational Linguistics, June 2011.

[141] Nguyen Vo and Kyumin Lee. Where are the facts? searching for fact-checked in-

formation to alleviate the spread of fake news. In *Conference on Empirical Methods in Natural Language Processing*, pages 7717–7731. Association for Computational Linguistics, November 2020.

[142] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes - which naive bayes? In *Conference on Email and Anti-Spam*, 2006.

[143] Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, volume 3201 of *Lecture Notes in Computer Science*, pages 217–226. Springer, 2004.

[144] Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913. Association for Computational Linguistics, 2021.

[145] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 2021.

[146] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*, 2017.

[147] Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. Isotropy in the contextual embedding space: Clusters and manifolds. In *International Conference on Learning Representations*, 2021.

[148] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR, 13–18 Jul 2020.

[149] Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. Should you mask 15% in masked language modeling? In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 2985–3000, 2023.

[150] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and

Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[151] Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. Pmi-masking: Principled masking of correlated spans. In *International Conference on Learning Representations*, 2021.

[152] Ankit Agrawal, Sarsij Tripathi, Manu Vardhan, Vikas Sihag, Gaurav Choudhary, and Nicola Dragoni. Bert-based transfer-learning approach for nested named-entity recognition using joint labeling. *Applied Sciences*, 12(3):976, 2022.

[153] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019.

[154] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

[155] Giuseppe Valenzise, Luigi Gerosa, Marco Tagliasacchi, Fabio Antonacci, and Augusto Sarti. Scream and gunshot detection and localization for audio-surveillance systems. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 21–26. IEEE Computer Society, 2007.

[156] Harsh Purohit, Ryo Tanabe, Takeshi Ichige, Takashi Endo, Yuki Nikaido, Kaori Suefusa, and Yohei Kawaguchi. MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection. In *Workshop on Detection and Classification of Acoustic Scenes and Events*, pages 209–213, 2019.

[157] Yuma Koizumi, Shoichiro Saito, Hisashi Uematsu, Noboru Harada, and Keisuke Imoto. Toyadmos: A dataset of miniature-machine operating sounds for anomalous sound detection. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2019.

[158] Yuma Koizumi, Yohei Kawaguchi, Keisuke Imoto, Toshiki Nakamura, Yuki Nikaido, Ryo Tanabe, Harsh Purohit, Kaori Suefusa, Takashi Endo, Masahiro Yasuda, and

Noboru Harada. Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring. In *Workshop on Detection and Classification of Acoustic Scenes and Events*, pages 81–85, 2020.

[159] Dan Stowell. Computational bioacoustics with deep learning: a review and roadmap. *PeerJ*, 10:e13152, 2022.

[160] Kyle Alspach. Does your boss sound a little funny? it might be an audio deepfake, 8 2022. `https://www.protocol.com/enterprise/deepfake-voice-cyberattack-ai-audio`; accessed January 19 2023.

[161] M Caldwell, JTA Andrews, T Tanay, and LD Griffin. Ai-enabled future crime. *Crime Science*, 9(1):1–13, 2020.

[162] Yisroel Mirsky, Ambra Demontis, Jaidip Kotak, Ram Shankar, Deng Gelei, Liu Yang, Xiangyu Zhang, Maura Pintor, Wenke Lee, Yuval Elovici, et al. The threat of offensive ai to organizations. *Computers & Security*, page 103006, 2022.

[163] Zhizheng Wu, Junichi Yamagishi, Tomi Kinnunen, Cemal Hanilçi, Md. Sahidullah, Aleksandr Sizov, Nicholas W. D. Evans, and Massimiliano Todisco. Asvspoof: The automatic speaker verification spoofing and countermeasures challenge. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):588–604, 2017.

[164] Andreas Nautsch, Xin Wang, Nicholas W. D. Evans, Tomi H. Kinnunen, Ville Vestman, Massimiliano Todisco, Héctor Delgado, Md. Sahidullah, Junichi Yamagishi, and Kong Aik Lee. Asvspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(2):252–265, 2021.

[165] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, et al. Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection. *arXiv preprint arXiv:2109.00537*, 2021.

[166] Dilrukshi Gamage, Jiayu Chen, Piyush Ghasiya, and Kazutoshi Sasahara. Deepfakes and society: What lies ahead? In *Frontiers in Fake Media Generation and Detection*, pages 3–43. Springer, 2022.

[167] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.

[168] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265. PMLR, 07–09 Jul 2015.

[169] Naser Damer, Alexandra Moseguí Saladié, Andreas Braun, and Arjan Kuijper. Morgan: Recognition vulnerability and attack detectability of face morphing attacks created by generative adversarial network. In *IEEE International Conference on Biometrics Theory, Applications and Systems*, pages 1–10, 2018.

[170] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K. Nayar. Face swapping: Automatically replacing faces in photographs. *ACM Transactions on Graphics*, 27(3):1–8, August 2008.

[171] Awais Khan, Khalid Mahmood Malik, James Ryan, and Mikul Saravanan. Voice spoofing countermeasures: Taxonomy, state-of-the-art, experimental analysis of generalizability, open challenges, and the way forward. *CoRR*, abs/2210.00417, 2022.

[172] Soubhik Barari, Christopher Lucas, and Kevin Munger. Political deepfakes are as credible as other fake media and (sometimes) real media, 1 2021. `osf.io/cdfh3`.

[173] Tomi Kinnunen, Md. Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas W. D. Evans, Junichi Yamagishi, and Kong-Aik Lee. The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection. In *Conference of the International Speech Communication Association*, pages 2–6. ISCA, 2017.

[174] Tao Zhang. Deepfake generation and detection, a survey. *Multimedia Tools and Applications*, 81(5):6259–6276, 2022.

[175] Markus Appel and Fabian Prietzel. The detection of political deepfakes. *Journal of Computer-Mediated Communication*, 27(4), 07 2022. zmac008.

[176] Sergi D Bray, Shane D Johnson, and Bennett Kleinberg. Testing human ability to

detect 'deepfake' images of human faces. *Journal of Cybersecurity*, 9(1):tyad011, 2023.

[177] Sophie J. Nightingale and Hany Farid. Ai-synthesized faces are indistinguishable from real faces and more trustworthy. *Proceedings of the National Academy of Sciences*, 119(8):e2120481119, 2022.

[178] Matthew Groh, Ziv Epstein, Chaz Firestone, and Rosalind Picard. Deepfake detection by human crowds, machines, and machine-informed crowds. *Proceedings of the National Academy of Sciences*, 119(1):e2110013119, 2022.

[179] Rashid Tahir, Brishna Batool, Hira Jamshed, Mahnoor Jameel, Mubashir Anwar, Faizan Ahmed, Muhammad Adeel Zaffar, and Muhammad Fareed Zaffar. Seeing is believing: Exploring perceptual differences in deepfake videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2021.

[180] Robert Geirhos, Roland S Zimmermann, Blair Bilodeau, Wieland Brendel, and Been Kim. Don't trust your eyes: on the (un) reliability of feature visualizations. In *International Conference on Machine Learning*, 2023.

[181] Gabrielle Watson, Zahra Khanjani, and Vandana P Janeja. Audio deepfake perceptions in college going populations. *arXiv preprint arXiv:2112.03351*, 2021.

[182] Xin Wang, Junichi Yamagishi, Massimiliano Todisco, Héctor Delgado, Andreas Nautsch, Nicholas Evans, Md Sahidullah, Ville Vestman, Tomi Kinnunen, Kong Aik Lee, et al. Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech. *Computer Speech & Language*, 64:101114, 2020.

[183] Nicolas M Müller, Karla Pizzi, and Jennifer Williams. Human perception of audio deepfakes. In *Workshop on Deepfake Detection for Audio Multimedia*, pages 85–91, 2022.

[184] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[185] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang,

and Cristian Canton Ferrer. The deepfake detection challenge dataset. *arXiv preprint arXiv:2006.07397*, 2020.

[186] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Conference on Computer Vision and Pattern Recognition*, pages 3207–3216, 2020.

[187] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *International Conference on Computer Vision*, pages 1–11, 2019.

[188] Kunlin Liu, Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Wenbo Zhou, and Weiming Zhang. Deepfacelab: Integrated, flexible and extensible face-swapping framework. *Pattern Recognition*, 141:109628, 2023.

[189] Kundan Kumar, Rithesh Kumar, Thibault De Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in Neural Information Processing Systems*, 32, 2019.

[190] Keith Ito and Linda Johnson. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[191] Databaker. Chinese standard mandarin speech corpus. `https://www.data-baker.com/open_source.html`, 2019.

[192] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. ESPnet: End-to-end speech processing toolkit. In *Conference of the International Speech Communication Association*, pages 2207–2211, 2018.

[193] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540, 2021.

[194] Xin Wang and Junichi Yamagishi. A comparative study on recent neural spoofing

countermeasures for synthetic speech detection. In *Conference of the International Speech Communication Association*, pages 4259–4263, 2021.

[195] Haoxin Ma, Jiangyan Yi, Chenglong Wang, Xinrui Yan, Jianhua Tao, Tao Wang, Shiming Wang, and Ruibo Fu. Cfad: A chinese dataset for fake audio detection. *Speech Communication*, 164:103122, 2024.

[196] Héctor Delgado, Nicholas Evans, Tomi Kinnunen, Kong Aik Lee, Xuechen Liu, Andreas Nautsch, Jose Patino, Md Sahidullah, Massimiliano Todisco, Xin Wang, and Others. ASVspoof 2021: Automatic speaker verification spoofing and countermeasures challenge evaluation plan. *arXiv preprint arXiv:2109.00535*, 2021.

[197] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[198] Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *Python in Science Conference*, 2010.

[199] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, USA, 2nd edition, 2014.

[200] Peggy P. K. Mok and Volker Dellwo. Comparing native and non-native speech rhythm using acoustic rhythmic measures: Cantonese, Beijing Mandarin and English. In *Proceedings of Speech Prosody*, pages 423–426, 2008.

[201] Mary Lee Hummert, Jaye L Shaner, Teri A Garstka, and Clark Henry. Communication with older adults: The influence of age stereotypes, context, and communicator age. *Human Communication Research*, 25(1):124–151, 1998.

[202] Elizabeth A Strand. Uncovering the role of gender stereotypes in speech perception. *Journal of Language and Social Psychology*, 18(1):86–100, 1999.

[203] Robert Chesney and Danielle Citron. Deepfakes and the new disinformation war: The coming age of post-truth geopolitics. *Foreign Affairs*, 98:147, 2019.

[204] Xuechen Liu, Xin Wang, Md. Sahidullah, Jose Patino, Héctor Delgado, Tomi Kinnunen, Massimiliano Todisco, Junichi Yamagishi, Nicholas W. D. Evans, Andreas Nautsch, and Kong Aik Lee. Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild. *IEEE ACM Trans. Audio Speech Lang. Process.*, 31:2507–2522, 2023.

[205] Rohit Arora, Aanchan Mohan, and Saket Anand. Impact of channel variation on one-class learning for spoof detection. *CoRR*, abs/2109.14900, 2021.

[206] Yang Xie, Zhenchuan Zhang, and Yingchun Yang. Siamese network with wav2vec feature for spoofing speech detection. In *Conference of the International Speech Communication Association*, pages 4269–4273. ISCA, 2021.

[207] Hemlata Tak, Massimiliano Todisco, Xin Wang, Jee-weon Jung, Junichi Yamagishi, and Nicholas W. D. Evans. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. In *Odyssey 2022: The Speaker and Language Recognition Workshop*, pages 112–119. ISCA, 2022.

[208] Jin Woo Lee, Eungbeom Kim, Junghyun Koo, and Kyogu Lee. Representation selective self-distillation and wav2vec 2.0 feature exploration for spoof-aware speaker verification. In *Conference of the International Speech Communication Association*. ISCA, 2022.

[209] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018.

[210] Kimberly T. Mai, Toby Davies, and Lewis D. Griffin. Self-supervised losses for one-class textual anomaly detection. *CoRR*, abs/2204.05695, 2022.

[211] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

[212] Jee-weon Jung, Hee-Soo Heo, Ju-ho Kim, Hye-jin Shim, and Ha-Jin Yu. Rawnet: Ad-

vanced end-to-end deep neural network using raw waveforms for text-independent speaker verification. In *Conference of the International Speech Communication Association*, pages 1268–1272. ISCA, 2019.

[213] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas W. D. Evans, and Anthony Larcher. End-to-end anti-spoofing with rawnet2. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6369–6373. IEEE, 2021.

[214] Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas W. D. Evans. AASIST: audio anti-spoofing using integrated spectro-temporal graph attention networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6367–6371. IEEE, 2022.

[215] Hemlata Tak, Massimiliano Todisco, Xin Wang, Jee-weon Jung, Junichi Yamagishi, and Nicholas W. D. Evans. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. *CoRR*, abs/2202.12233, 2022.

[216] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE ACM Transactions on Audio Speech and Language Processing*, 29:3451–3460, 2021.

[217] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. In *Conference of the International Speech Communication Association*, pages 3465–3469. ISCA, 2019.

[218] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, 2020.

[219] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[220] You Zhang, Fei Jiang, and Zhiyao Duan. One-class learning towards synthetic voice spoofing detection. *IEEE Signal Processing Letters*, 28:937–941, 2021.

[221] Siwen Ding, You Zhang, and Zhiyao Duan. Samo: Speaker attractor multi-center

one-class learning for voice anti-spoofing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5. IEEE, 2023.

[222] Alessandro Pianese, Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Deepfake audio detection by speaker verification. In *IEEE International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2022.

[223] Federico Alegre, Asmaa Amehraye, and Nicholas W. D. Evans. A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns. In *IEEE Conference on Biometrics: Theory, Applications and Systems*, pages 1–8. IEEE, 2013.

[224] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[225] Jesús Antonio Villalba López, Antonio Miguel, Alfonso Ortega, and Eduardo Lleida. Spoofing detection with DNN and one-class SVM for the asvspoof 2015 challenge. In *Conference of the International Speech Communication Association*, pages 2067–2071. ISCA, 2015.

[226] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18699–18708. IEEE, 2022.

[227] Hasam Khalid and Simon S. Woo. Oc-fakedect: Classifying deepfakes using one-class variational autoencoder. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2794–2803, 2020.

[228] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *IEEE/CVF International Conference on Computer Vision*, pages 1705–1714. IEEE, 2019.

[229] Chao Feng, Ziyang Chen, and Andrew Owens. Self-supervised video forensics by audio-visual anomaly detection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[230] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Why normalizing flows fail to detect out-of-distribution data. In *Advances in Neural Information Processing Systems*, 2020.

[231] Lily H. Zhang, Mark Goldstein, and Rajesh Ranganath. Understanding failures in out-of-distribution detection with deep generative models. In *International Conference on Machine Learning*, volume 139, pages 12427–12436. PMLR, 2021.

[232] Davide Cozzolino, Andreas Rössler, Justus Thies, Matthias Nießner, and Luisa Verdoliva. Id-reveal: Identity-aware deepfake video detection. In *IEEE/CVF International Conference on Computer Vision*, pages 15088–15097. IEEE, 2021.

[233] Davide Cozzolino, Alessandro Pianese, Matthias Nießner, and Luisa Verdoliva. Audio-visual person-of-interest deepfake detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 943–952, 2023.

[234] Alexandros Haliassos, Rodrigo Mira, Stavros Petridis, and Maja Pantic. Leveraging real talking faces via self-supervision for robust forgery detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14930–14942. IEEE, 2022.

[235] Liang Shi, Jie Zhang, and Shiguang Shan. Real face foundation representation learning for generalized deepfake detection. *CoRR*, abs/2303.08439, 2023.

[236] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: a simple framework for masked image modeling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9643–9653. IEEE, 2022.

[237] Penny Chong, Lukas Ruff, Marius Kloft, and Alexander Binder. Simple and effective prevention of mode collapse in deep one-class classification. In *International Joint Conference on Neural Networks*, pages 1–9. IEEE, 2020.

[238] Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. https://doi.org/10.7488/ds/2645.

[239] Joel Frank and Lea Schönherr. Wavefake: A data set to facilitate audio deepfake de-

tection. In *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

[240] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6199–6203. IEEE, 2020.

[241] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems*, 2020.

[242] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3617–3621. IEEE, 2019.

[243] Ryosuke Sonobe, Shinnosuke Takamichi, and Hiroshi Saruwatari. JSUT corpus: free large-scale japanese speech corpus for end-to-end speech synthesis. *CoRR*, abs/1711.00354, 2017.

[244] Zhenyu Zhang, Yewei Gu, Xiaowei Yi, and Xianfeng Zhao. FMFCC-A: A challenging mandarin dataset for synthetic speech detection. In *International Workshop on Digital Forensics and Watermarking*, volume 13180 of *Lecture Notes in Computer Science*, pages 117–131. Springer, 2021.

[245] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. AISHELL-1: an open-source mandarin speech corpus and a speech recognition baseline. In *Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech Systems and Assessment*, pages 1–5. IEEE, 2017.

[246] Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. AISHELL-3: A multi-speaker mandarin TTS corpus. In *Conference of the International Speech Communication Association*, pages 2756–2760. ISCA, 2021.

[247] Dong Wang and Xuewei Zhang. THCHS-30 : A free chinese speech corpus. *CoRR*, abs/1512.01882, 2015.

[248] Zehui Yang, Yifan Chen, Lei Luo, Runyan Yang, Lingxuan Ye, Gaofeng Cheng, Ji Xu,

Yaohui Jin, Qingqing Zhang, Pengyuan Zhang, Lei Xie, and Yonghong Yan. Open source magicdata-ramc: A rich annotated mandarin conversational(ramc) speech dataset. In *Conference of the International Speech Communication Association*, pages 1736–1740. ISCA, 2022.

[249] Yuan Gong, Yu-An Chung, and James R. Glass. AST: audio spectrogram transformer. In *Conference of the International Speech Communication Association*, pages 571–575. ISCA, 2021.

[250] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5206–5210. IEEE, 2015.

[251] Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, Di Wu, and Zhendong Peng. WENETSPEECH: A 10000+ hours multi-domain mandarin corpus for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6182–6186. IEEE, 2022.

[252] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 776–780. IEEE, 2017.

[253] Yuan Gong, Cheng-I Lai, Yu-An Chung, and James R. Glass. SSAST: self-supervised audio spectrogram transformer. In *AAAI Conference on Artificial Intelligence*, pages 10699–10709. AAAI Press, 2022.

[254] Min Xu, Ling-Yu Duan, Jianfei Cai, Liang-Tien Chia, Changsheng Xu, and Qi Tian. Hmm-based audio keyword generation. In *Pacific Rim Conference on Multimedia*, volume 3333 of *Lecture Notes in Computer Science*, pages 566–574. Springer, 2004.

[255] Xinhui Zhou, Daniel Garcia-Romero, Ramani Duraiswami, Carol Y. Espy-Wilson, and Shihab A. Shamma. Linear versus mel frequency cepstral coefficients for speaker recognition. In *IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 559–564. IEEE, 2011.

[256] Yao-Yuan Yang, Moto Hira, Zhaoheng Ni, Artyom Astafurov, Caroline Chen, Chris-

tian Puhrsch, David Pollack, Dmitriy Genzel, Donny Greenberg, Edward Z Yang, et al. Torchaudio: Building blocks for audio and speech processing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6982–6986. IEEE, 2022.

[257] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *International Conference on Learning Representations*, 2022.

[258] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8065–8081, 2022.

[259] Zhiyun Fan, Meng Li, Shiyu Zhou, and Bo Xu. Exploring wav2vec 2.0 on speaker verification and language identification. In *Conference of the International Speech Communication Association*, pages 1509–1513. ISCA, 2021.

[260] Xiaofei Ma, Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. Universal text representation from bert: An empirical study. *arXiv preprint arXiv:1910.07973*, 2019.

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

[262] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4911–4920. IEEE, 2022.

[263] Kevin P. Murphy. *Machine learning - a probabilistic perspective.* Adaptive computation and machine learning series. MIT Press, 2012.

[264] Hemlata Tak, Madhu R. Kamble, Jose Patino, Massimiliano Todisco, and Nicholas W. D. Evans. Rawboost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6382–6386. IEEE, 2022.

[265] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. Specaugment: A simple data augmentation method for

automatic speech recognition. In Gernot Kubin and Zdravko Kacic, editors, *Conference of the International Speech Communication Association*, pages 2613–2617. ISCA, 2019.

[266] Daniel W. Griffin and Jae S. Lim. Signal estimation from modified short-time fourier transform. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 804–807. IEEE, 1983.

[267] Kimberly T Mai, Toby Davies, and Lewis D Griffin. Understanding the limitations of self-supervised learning for tabular anomaly detection. *Pattern Analysis and Applications*, 27(2):61, 2024.

[268] Tom Shenkar and Lior Wolf. Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*, 2022.

[269] Shebuti Rayana. Odds library, 2016. `https://odds.cs.stonybrook.edu`.

[270] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. In *NeurIPS*, 2022.

[271] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[272] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2021.

[273] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems*, 2022.

[274] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. VIME: extending the success of self- and semi-supervised learning to tabular domain. In *Advances in Neural Information Processing Systems*, 2020.

[275] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting

deep learning models for tabular data. In *Advances in Neural Information Processing Systems*, pages 18932–18943, 2021.

[276] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE Computer Society, 2016.

[277] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):5962–5979, 2022.

[278] Georg Steinbuss and Klemens Böhm. Benchmarking unsupervised outlier detection with realistic synthetic data. *ACM Transactions on Knowledge Discovery from Data*, 15(4):65:1–65:20, 2021.

[279] Thomas Nagler, Daniel Krüger, and Aleksey Min. Stationary vine copula models for multivariate time series. *Journal of Econometrics*, 227(2):305–324, 2022.

[280] Salvatore Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Philip Chan. KDD Cup 1999 Data. UCI Machine Learning Repository, 1999. `https://doi.org/10.24432/C51C7N`.

[281] Ryo Kamoi and Kei Kobayashi. Why is the mahalanobis distance effective for anomaly detection? *CoRR*, abs/2003.00402, 2020.

[282] Bernhard Schölkopf, Robert C. Williamson, Alexander J. Smola, John Shawe-Taylor, and John C. Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, pages 582–588, 1999.

[283] Kuang-Huei Lee, Anurag Arnab, Sergio Guadarrama, John F. Canny, and Ian Fischer. Compressive visual representations. In *Advances in Neural Information Processing Systems*, pages 19538–19552, 2021.

[284] Ravid Shwartz Ziv and Yann LeCun. To compress or not to compress—self-supervised learning and information theory: A review. *Entropy*, 26(3):252, 2024.

[285] Yaodong Yu, Sam Buchanan, Druv Pai, Tianzhe Chu, Ziyang Wu, Shengbang Tong, Hao Bai, Yuexiang Zhai, Benjamin D Haeffele, and Yi Ma. White-box trans-

formers via sparse rate reduction: Compression is all there is? *arXiv preprint arXiv:2311.13110*, 2023.

[286] Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. Gpt detectors are biased against non-native english writers. *Patterns*, 4(7):100779, 2023.

[287] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91. PMLR, 2018.

[288] United Kingdom. Article 22, UK General Data Protection Regulation (UK GDPR). `https://www.legislation.gov.uk/eur/2016/679`, 2018. Retained EU Regulation 2016/679 under the Data Protection Act 2018.

[289] Jiangyan Yi, Ruibo Fu, Jianhua Tao, Shuai Nie, Haoxin Ma, Chenglong Wang, Tao Wang, Zhengkun Tian, Ye Bai, Cunhang Fan, Shan Liang, Shiming Wang, Shuai Zhang, Xinrui Yan, Le Xu, Zhengqi Wen, and Haizhou Li. ADD 2022: the first audio deep synthesis detection challenge. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 9216–9220. IEEE, 2022.

[290] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *International Conference on Machine Learning*, volume 139, pages 8748–8763. PMLR, 2021.

[291] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, volume 162, pages 1298–1312. PMLR, 2022.

[292] C2PA. C2pa technical specification. Technical report, C2PA, 2021.

[293] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[294] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning:

A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[295] Ling Huang, XuanLong Nguyen, Minos N. Garofalakis, Michael I. Jordan, Anthony D. Joseph, and Nina Taft. In-network PCA and anomaly detection. In *Advances in Neural Information Processing Systems*, pages 617–624, 2006.

[296] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: identifying density-based local outliers. In *ACM International Conference on Management of Data*, pages 93–104. ACM, 2000.

[297] Sreekanth Vempati, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Generalized RBF feature maps for efficient detection. In *British Machine Vision Conference*, pages 1–11, 2010.

[298] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[299] Jie Ren, Stanislav Fort, Jeremiah Z. Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *CoRR*, abs/2106.09022, 2021.

[300] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

[301] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8759–8773. PMLR, 2022.

[302] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin W. Wilson. CNN architectures for large-scale audio classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 131–135. IEEE, 2017.

[303] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and D. Mike Titterington, editors, *nternational Conference on Artificial Intelligence and Statistics*, volume 9 of *JMLR Proceedings*, pages 249–256, 2010.

[304] Paul Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern Recognition and Artificial Intelligence*, 116:374–388, 1976.