Beyond Individual Input for Deep Anomaly Detection on Tabular Data

Hugo Thimonier **Fabrice Popineau** **Arpad Rimmel** **Bich-Liên Doan**

Université Paris-Saclay, CNRS, CentraleSupélec,
Laboratoire Interdisciplinaire des Sciences du Numérique,
91190, Gif-sur-Yvette, France.
name.surname@lisn.fr

Abstract

Anomaly detection is vital in many domains, such as finance, healthcare, and cybersecurity. In this paper, we propose a novel deep anomaly detection method for tabular data that leverages Non-Parametric Transformers (NPTs), a model initially proposed for supervised tasks, to capture both feature-feature and sample-sample dependencies. In a reconstruction-based framework, we train the NPT to reconstruct masked features of normal samples. In a non-parametric fashion, we leverage the whole training set during inference and use the model's ability to reconstruct the masked features to generate an anomaly score. To the best of our knowledge, this is the first work to successfully combine feature-feature and sample-sample dependencies for anomaly detection on tabular datasets. Through extensive experiments on 31 benchmark tabular datasets, we demonstrate that our method achieves state-of-the-art performance, outperforming existing methods by 2.4% and 1.2% in terms of F1-score and AUROC, respectively. Our ablation study provides evidence that modeling both types of dependencies is crucial for anomaly detection on tabular data.

1 Introduction

Anomaly detection is a critical task that aims to identify samples that deviate from a pre-defined notion of normality within a dataset. Traditional approaches to anomaly detection characterize the *normal*[1] distribution almost exclusively using samples considered as *normal*, and flag data points as anomalies based on their deviation from this distribution. Anomaly detection (AD) is especially useful for applications involving imbalanced datasets, where standard supervised methods may fail to achieve satisfactory performance [52]. Those applications include fraud detection [18], intrusion detection in cybersecurity [27], astronomy [35], medical diagnosis [6], and data cleaning to remove samples that may hinder the performance of machine learning models.

Anomaly detection encompasses both unsupervised and supervised methods. In most real-world scenarios, labeled datasets that differentiate normal samples from anomalies are unavailable or costly to obtain. To address this, efficient anomaly detection methods must be robust to dataset contamination, where the training set is predominantly composed of normal samples but also includes anomalies. However, when labeled data is available, one can consider a supervised approach to create a training set consisting solely of *normal* samples, thereby indirectly incorporating label information into the anomaly detection model.

Many general AD methods tend to work well on tasks that involve unstructured data (*e.g.*, natural language processing or computer vision) such as [41, 48, 25, 37, 38, 21, 26]. However, recent work

[1]The term *normal* here relates to the concept of normality in opposition to *abnormal*.

Preprint. Under review.

[4, 32, 43] has revealed that the best-performing methods for tabular data involve models tailored to consider the particular structure of this data type. Anomaly detection methods for structured data typically use either *feature-feature* or *sample-sample* dependencies to identify anomalies. For instance, in [43], the authors assume a class-dependent relationship between a subset of variables in a sample's feature vector and the rest of its variables. The authors thus propose a contrastive learning framework to detect anomalies based on this assumption. Another recent method [49] identifies anomalies in tabular datasets by focusing on sample-sample dependencies. This approach uses a variational autoencoder to estimate the *normal* distribution and subsequently computes the influence of *normal* samples on validation samples to construct an anomaly score. Both approaches have demonstrated competitive results for anomaly detection in tabular datasets.

Recent work on supervised deep learning methods for tabular data [42, 1, 11, 47, 23] has also highlighted the importance of considering the particular structure of tabular data. In particular, in [23, 47], the authors emphasize the significance of considering both feature-feature and sample-sample dependencies for supervised regression and classification problems on tabular data. Based on the latter observation, we formulate the hypothesis that not only are feature-feature relations class-dependent as supported by [43] but **sample-sample dependencies are also class-dependent** and can be used to identify anomalies. In particular, since interactions between samples are learned exclusively using *normal* samples in the anomaly detection setup, they should be especially discriminative in identifying anomalies during inference.

To test this hypothesis, we employ Non-Parametric Transformers (NPT) [23], first proposed for supervised tasks on tabular datasets. NPTs leverage two attention mechanisms to capture these relations between samples and between features: Attention Between Datapoints (ABD) and Attention Between Attributes (ABA). We show that NPTs are particularly relevant for flagging anomalies, in line with recent work [15] demonstrating the effectiveness of new deep learning architectures, such as transformers [50], for anomaly detection on tabular data. We experiment on an extensive benchmark of tabular datasets to demonstrate the capacity of our approach to detect anomalies and compare our performances to existing AD methods. We obtain state-of-the-art results when it comes to detection accuracy. We also test the robustness of our approach to dataset contamination and give evidence that it can serve for unsupervised anomaly detection when the training set contamination is not too severe. Finally, our ablation study, conducted with a reconstruction-based approach similar to our proposed method but utilizing K-nearest neighbors (KNN) imputation, provides evidence that considering both types of dependencies can be crucial to accurately detect anomalies on specific datasets.

The present work offers the following contributions:

- We put forward the first deep anomaly detection method to successfully combine feature-feature and sample-sample dependencies.
- Our method shows state-of-the-art anomaly detection capacity on an extensive benchmark of 31 tabular datasets.
- Our reconstruction-based method shows robustness to small data contamination.
- We provide evidence of the crucial role of considering both dependencies for anomaly detection on tabular data.

2 Related works

Anomaly detection approaches can be categorized into four main types: density estimation, one-class classification, reconstruction-based, and self-supervised.

Density Estimation The most straightforward approach to detecting samples that do not belong to a distribution is to estimate the distribution directly and to measure the likelihood of a sample under the estimated distribution. Several approaches found in the literature have considered using non-parametric density estimation methods to estimate the density of the *normal* distribution, such as KDE [30], GMM [36], or Copula as in COPOD [24]. Other approaches also focused on local density estimation to detect outliers, such as Local Outlier Factor (LOF) [5]. In inference, one flags as anomalies the samples that lie in low-probability regions under the estimated distribution.

Reconstruction Based Methods Other methods have consisted in learning to reconstruct samples that belong to the *normal* distribution. In this framework, the models' incapacity to reconstruct a

sample correctly serves as a proxy to measure anomaly. A high reconstruction error would indicate that a sample does not belong to the estimated *normal* distribution. Those approaches can involve PCA [17] or neural networks such as diverse types of autoencoders [51, 31, 6, 21], or GANs [39, 40].

One-Class Classification The term *one-class classification* (OCC) was coined in [28] and describes identifying anomalies without directly estimating the *normal* density. One-class classification involves discriminative models which directly estimate a decision boundary. For instance, in kernel-based approaches [41, 48], authors propose to characterize the support of the *normal* samples in a Hilbert space and to flag as anomalies the samples that would lie outside of the estimated support. Similarly, recent work has extended their approach by replacing kernels with deep neural networks [37]. In the latter approach, neural networks must be constrained in their architectures to avoid model collapse, *i.e.* mapping all *normal* samples to a single value when minimizing a one-class loss. Thus, in [7], authors proposed regularization techniques to alleviate this issue. In [12], authors proposed DROCC that involves generating, in the course of training, synthetic anomalous samples in order to learn a classifier on top of the one-class representation. Other OCC approaches have relied on tree-based model such as isolation forest (IForest) [25], extended isolation forest [16], RRCF [14] and PIDForest [10].

Self-Supervised Approaches Recent methods have also considered self-supervision as a means to identify anomalies. In [4], authors apply several affine transformations to each sample and train a classifier to identify from the transformed samples which transformation was applied. The classifier only learns to discriminate between transformations using *normal* transformed samples: assuming this problem is class-dependent, the classifier should fail to identify transformation applied to anomalies. In [32], authors propose a contrastive framework in which samples are transformed using neural mappings and are embedded in a latent semantic space using an encoder. The objective is to learn transformations so that transformed samples still share similarities with their untransformed counterpart while different transformations are easily distinguishable. The contrastive loss then serves as the anomaly score in inference. Similarly, [43] also propose a contrastive framework in which they identify samples as anomalies based on their inter-feature relations. Other self-supervised approaches, such as [46, 34], have focused on representation learning to foster the performance of one-class classification models.

Attention Mechanisms First introduced in [50], the concept of attention has become ubiquitous in the machine learning literature. Scholars have successfully applied transformers on a broad range of tasks, including computer vision, *e.g.* image generation with the Image Transformer [29] or image classification with the Vision Transformer (ViT) [9], natural language processing *e.g.* Masked Language Models (MLM) such as BERT [8], and classification tasks on structured datasets [47, 23].

Deep Learning for Tabular Data Despite the effectiveness of deep learning models for numerous tasks involving unstructured data, non-deep models remain the prevalent choice for machine learning tasks such as classification and regression on tabular data [13, 44]. However, in recent years scholars have shown that one could successfully resort to deep learning methods for various tasks on tabular datasets. For instance, in [42, 19], authors discuss how regularization is crucial in training a deep learning model tailored for tabular data. Hence, they propose a new regularization loss to accommodate the variability between features. Similarly, [20] shows that correctly selecting a combination of regularization techniques can suffice for a Multi-Layer Perceptron (MLP) to compete with GBDT. Finally, [47, 23] propose deep learning models based on attention mechanisms that rely on feature-feature, feature-label, sample-sample, and sample-label attention. Both models achieve competitive results on several baseline datasets and emphasize sample-sample interaction's role in classifying samples correctly.

3 Method

In this section, we discuss the learning objective used to optimize the parameters of our model, then we briefly present the mechanisms involved in Non-Parametric Transformers [23], the core model used in our approach, and finally, we present NPT-AD, our method to derive an anomaly score.

3.1 Learning Objective

Reconstruction-based approaches for anomaly detection involve training a model to accurately reconstruct *normal* samples while failing to reconstruct anomaly samples. Such methods effectively identify anomalies by exploiting differences in the underlying data distributions between *normal* and anomalous samples. Let $\mathcal{D}_{train} = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^n$ represent the training set composed of n *normal* samples with d features. Standard reconstruction-based approaches consider the task of learning a mapping $\phi_\theta : \mathbb{R}^d \to \mathbb{R}^d$ to minimize a reconstruction loss. The parameters $\theta \in \Theta$ are optimized to reconstruct each sample $\mathbf{x} \in \mathbb{R}^d$ in the training set with minimal error. Formally, the overall objective can be expressed as

$$\min_{\theta \in \Theta} \sum_{\mathbf{x} \in \mathcal{D}_{train}} d(\mathbf{x}, \phi_\theta(\mathbf{x})), \tag{1}$$

where $d(\mathbf{x}, \phi_\theta(\mathbf{x}))$ measures how well the model reconstructs sample $\mathbf{x}$. The latter is often set to be a distance measure such as the Euclidean distance.

The AD method proposed in [43] employs a masking strategy that maximizes the mutual information between each sample and its masked-out part by minimizing a contrastive loss. Recently, [22] demonstrated how stochastic masking [8] also maximizes mutual information, thereby establishing a link between the method of [43] and stochastic masking. In stochastic masking, each entry in a sample vector $\mathbf{x} \in \mathbb{R}^d$ is masked with probability p_{mask}, and the objective task is to predict the masked-out features from the unmasked features. Formally, let $m \in \mathbb{R}^d$ be a binary vector taking value 1 when the corresponding entry in $\mathbf{x}$ is masked, $\mathbf{x}^m = \{x_j : m_j = 1\}$ represents the masked entries of sample $\mathbf{x}$, and $\mathbf{x}^o = \{x_j : m_j = 0\}$ denotes the complement of $\mathbf{x}^m$, composed of the observed features of sample $\mathbf{x}$. In this framework, the objective in eq. 1 is modified to

$$\min_{\theta \in \Theta} \sum_{\mathbf{x} \in \mathcal{D}_{train}} d(\mathbf{x}^m, \phi_\theta(\mathbf{x}^o)), \tag{2}$$

where $\phi_\theta(\mathbf{x}^o)$ denotes the reconstructed masked features of sample $\mathbf{x}$ by the model.

Our proposed approach leverages the entire dataset in a non-parametric manner to reconstruct masked features. This method considers feature-feature interactions and also captures relationships between samples to optimize the reconstruction objective. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ denote the dataset matrix, consisting of n training samples with d features. We introduce the matrix equivalents of m, $\mathbf{x}^m$, and $\mathbf{x}^o$, denoted as $\mathbf{M}$, $\mathbf{X}^M$, and $\mathbf{X}^O$, respectively, all in $\mathbb{R}^{n \times d}$. The reconstruction objective described in eq. 2 can then be reformulated as

$$\min_{\theta \in \Theta} \sum_{\mathbf{x} \in \mathcal{D}_{train}} d\left(\mathbf{x}^m, \phi_\theta\left(\mathbf{x}^o \mid \mathbf{X}^O\right)\right). \tag{3}$$

3.2 Non-parametric transformer (NPT)

We resort to Non-Parametric Transformer (NPT) [23] as the core model for our approach, denoted as ϕ_θ in section 3.1. NPT involves both attention between features and attention between samples, thus allowing the ability to capture feature-feature and sample-sample dependencies. More precisely, two mechanisms involved in NPTs allow anomalies to be identified: Attention Between Datapoints (ABD) and Attention Between Attributes (ABA). Both attention mechanisms rely on multi-head self-attention (MHSA), which was first introduced in the natural-language processing literature [3, 8, 50]. We discuss MHSA more thoroughly in App. A and only detail in this section the two mechanisms put forward in [23].

As an input, NPT receives both the dataset and a masking matrix $(\mathbf{X}, \mathbf{M}) \in \mathbb{R}^{n \times d} \times \mathbb{R}^{n \times d}$. Before feeding the input to the NPT, we pass each of the n data samples through a linear embedding layer to obtain an e-dimensional embedding for each feature. Thus, as an input, NPT receives a representation $\mathbf{H}^0 \in \mathbb{R}^{n \times d \times e}$. A sequence of MHSA layers is applied to the input, alternating between ABA and ABD. The model then outputs a prediction for masked features while keeping unmasked features unchanged $\widehat{\mathbf{X}} \in \mathbb{R}^{n \times d}$.

Attention Between Datapoints (ABD) It is the key feature that differentiates NPT from standard transformer models. This mechanism captures **pairwise relation between data samples**. Consider as an input to the ABD layer the previous layer representation $\mathbf{H}^{(\ell)} \in \mathbb{R}^{n \times d \times e}$ flattened to $\mathbb{R}^{n \times h}$

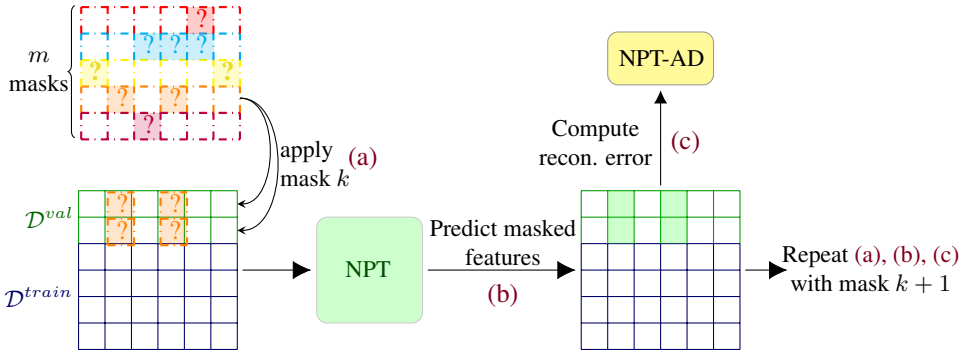


Figure 1: NPT-AD inference pipeline. In step (a), mask j is applied to each validation sample. We construct a matrix $\mathbf{X}$ composed of the masked validation samples and the whole *unmasked* training set. In step (b), we feed $\mathbf{X}$ to the Non-Parametric Transformer (NPT), which tries to reconstruct the masked features for each validation sample. On top of the learned feature-feature interactions, NPT will use the unmasked training samples to reconstruct the mask features. In step (c), we compute the reconstruction error that we later aggregate in the NPT-AD score.

where $h = d \cdot e$. Then, NPT applies MHSA, as seen in equation 12 in appendix A, between the data samples flattened representations $\{\mathbf{H}_i^{(\ell)} \in \mathbb{R}^{1\times h} | i \in 1, \dots, n\}$.

$$\text{ABD}(\mathbf{H}^{(\ell)}) = \text{MHSA}(\mathbf{H}^{(\ell)}) = \mathbf{H}^{(\ell+1)} \in \mathbb{R}^{n\times h} \tag{4}$$

After applying ABD, the data representation is reshaped to its original dimension in $\mathbb{R}^{n\times d\times e}$.

Attention Between Attributes (ABA) As already discussed, NPT alternates between ABD and ABA layers. ABA layers should help learn per data sample representation for the inter-sample representations. In contrast with ABD, ABA consists in applying MHSA independently to each row in $\mathbf{H}^{(\ell)}$, *i.e.* to each data sample's intermediate representation $\mathbf{H}_i^{(\ell)} \in \mathbb{R}^{d\times e}, i \in \{1, \dots, n\}$.

$$\text{ABA}(\mathbf{H}^{(\ell)}) = \underset{\text{axis=n}}{\text{stack}} \left(\text{MHSA}(\mathbf{H}_1^{(\ell)}), \dots, \text{MHSA}(\mathbf{H}_n^{(\ell)})\right) \in \mathbb{R}^{n\times d\times e} \tag{5}$$

3.3 Anomaly score

We directly derive the anomaly score from the loss optimized during training. For numerical features, the loss corresponds to the squared difference between the reconstructed feature and its actual value. Meanwhile, for categorical features, we use the cross-entropy loss function. The anomaly score relies on our model's capacity to reconstruct masked features correctly and assumes that the model should better reconstruct *normal* samples. Two reasons support this assumption. First, relations between features are class-dependent, as supported by [43]; having observed only *normal* samples in the training phase, the model should be unable to fetch the learned feature-feature interactions to reconstruct anomalies properly. Second, sample-sample interactions seen by the model only correspond to interactions between *normal* samples, making it difficult to successfully exploit interactions between *normal* samples and anomalies.

As detailed in Figure 1, we consider m d-dimensional deterministic mask vectors that designate which of the d features of *each* validation sample will be hidden. We set the maximum number of features to be masked simultaneously r, and construct $m = \sum_{k=1}^{r} \binom{d}{k}$ masks. Each mask is applied to each validation sample $\mathbf{z} \in \mathcal{D}^{val}$ to obtain m different masked samples $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ of the original sample $\mathbf{z}$. We use the whole unmasked training set[2] $\mathcal{D}^{train}$ to predict the masked features of each sample for each of the m masked vectors and construct the anomaly score for a validation sample $\mathbf{z}$ as

$$\text{NPT-AD}(\mathbf{z}; \mathcal{D}^{train}) = \frac{1}{m} \sum_{k=1}^{m} \mathcal{L}_{features}(\mathbf{z}^{(k)}; \mathcal{D}^{train}), \tag{6}$$

[2]For large datasets, we resort to a random subsample of the training set for computational reasons.

Table 1: Deep models: anomaly detection F1-score (↑). We perform 5% T-test to test whether the difference between the highest metrics for each dataset is statistically significant.

Method	DROCC (abalone)	GOAD (thyroid)	NeuTraL-AD (arrhy.)	Internal Cont.	NPT-AD
Wine	63.0±20.0	67.0±9.4	78.2±4.5	**90.0**±6.3	72.5±7.7
Lympho	65.0±5.0	68.3±13.0	20.0±18.7	86.7±6.0	**94.2**±7.9
Glass	14.5±11.1	12.7±3.9	9.0±4.4	**27.2**±10.6	**26.2**±10.9
Vertebral	**9.3**±6.1	**16.3**±9.6	3.8±1.2	**26.0**±7.7	20.3±4.8
Wbc	9.0±6.2	**66.2**±2.9	60.9±5.6	**67.6**±3.6	**67.3**±1.7
Ecoli	N/A	61.4±31.7	7.0±7.1	70.0±7.8	**77.7**±0.1
Ionosph.	76.9±2.8	83.4±2.6	90.6±2.4	**93.2**±1.3	**92.7**±0.6
Arrhyth.	37.1±6.8	52.0±2.3	59.5±2.6	**61.8**±1.8	60.4±1.4
Breastw	93.0±3.7	**96.0**±0.6	91.8±1.3	**96.1**±0.7	95.7±0.3
Pima	66.0±4.1	66.0±3.1	60.3±1.4	59.1±2.2	**68.8**±0.6
Vowels	66.2±8.8	31.1±4.2	10.0±6.2	**90.8**±1.6	88.7±1.6
Letter	55.6±3.6	20.7±1.7	5.7±0.8	62.8±2.4	**71.4**±1.9
Cardio	49.8±3.2	**78.6**±2.5	45.5±4.3	71.0±2.4	**78.1**±0.1
Seismic	19.1±0.9	24.1±1.0	11.8±4.3	20.7±1.9	**26.2**±0.7
Musk	99.4±1.5	**100.0**±0.0	99.0±0.0	**100.0**±0.0	**100.0**±0.0
Speech	4.3±2.0	4.8±2.3	4.7±1.4	5.2±1.2	**9.3**±0.8
Thyroid	72.7±3.1	72.5±2.8	69.4±1.4	76.8±1.2	**77.0**±0.6
Abalone	17.9±1.3	57.6±2.2	53.2±4.0	**68.7**±2.3	59.7±0.1
Optdigits	30.5±5.2	0.3±0.3	16.2±7.3	**66.3**±10.1	**62.0**±2.7
Satimage2	4.8±1.6	90.7±0.7	92.3±1.9	92.4±0.7	**94.8**±0.8
Satellite	52.2±1.5	64.2±0.8	71.6±0.6	73.2±1.6	**74.6**±0.7
Pendigits	11.0±2.6	40.1±5.0	69.8±8.7	82.3±4.5	**92.5**±1.3
Annthyr.	64.2±3.3	50.3±6.3	44.1±2.3	45.4±1.8	57.7±0.6
Mnist	N/A	66.9±1.3	84.8±0.5	85.9±0.0	**71.8**±0.3
Mammo.	32.6±2.1	33.7±6.1	19.2±2.4	29.4±1.4	**43.6**±0.5
Shuttle	N/A	73.5±5.1	97.9±0.2	**98.4**±0.1	**98.2**±0.3
Mullcross	N/A	99.7±0.8	96.3±10.5	**100.0**±0	**100.0**±0
Forest	N/A	0.1±0.2	51.6±8.2	44.0±4.1	**58.0**±10
Campaign	N/A	16.2±1.8	42.1±1.7	46.8±1.4	**49.8**±0.3
Fraud	N/A	53.1±10.2	24.3±7.8	**57.9**±2.8	**58.1**±3.2
Backdoor	N/A	12.7±2.9	84.4±1.8	**86.6**±0.1	84.1±0.1
mean	32.7	51.0	50.8	67.2	**68.8**
mean std	3.4	4.4	4.0	2.9	**2.0**
mean rank	10.8	7.8	9.0	3.5	**3.0**

where $\mathcal{L}_{features}(\mathbf{z}^{(k)}; \mathcal{D}^{train})$ designates the loss for the sample $\mathbf{z}$ with mask k. We also considered other forms of aggregation, such as the maximum loss over all masks.

4 Experiments

Datasets We experiment on an extensive benchmark of tabular datasets following previous work [43]. The benchmark is comprised of two datasets widely used in the anomaly detection literature, namely Arrhythmia and Thyroid, a second group of datasets, the "Multi-dimensional point datasets", obtained from the Outlier Detection DataSets (ODDS)[3] containing 28 datasets. We omit the datasets Heart and Yeast following previous work [43] and also omit the KDD dataset since it presents a certain number of limitations [45]. Instead, we include three real-world datasets from [15] that display relatively similar characteristics to KDD in terms of dimensions: fraud, campaign and backdoor. See App. B for more detail on the datasets' characteristics.

Experimental settings Per the literature [55, 4], we construct the training set with a random subsample of the *normal* samples representing 50% of the *normal* samples, we concatenate the 50% remaining with the entire set of anomalies to constitute the validation set. Following previous work, [4, 43], the decision threshold for the NPT-AD score is chosen such that the number of predicted

[3] http://odds.cs.stonybrook.edu/

Table 2: Non-deep models: anomaly detection F1-score (↑). We perform 5% T-test to test whether the difference between the highest metrics for each dataset is statistically significant.

Method	COPOD	IForest	KNN	PIDForest	RRCF	NPT-AD
Wine	60.0±4.5	64.0±12.8	**94.0**±4.9	50.0±6.4	69.0±11.4	72.5±7.3
Lympho	85.0±5.0	71.7±7.6	80.0±11.7	70.0±0.0	36.7±18.0	**94.2**±7.9
Glass	11.1±0.0	11.1±0.0	11.1±9.7	8.9±6.0	15.6±13.3	**26.2**±10.9
Vertebral	1.7±1.7	13.0±3.8	10.0±4.5	12.0±5.2	8.0±4.8	**20.3**±4.8
Wbc	**71.4**±0.0	70.0±3.7	63.8±2.3	65.7±3.7	54.8±6.1	67.3±1.7
Ecoli	25.6±11.2	58.9±22.2	**77.8**±3.3	25.6±11.2	28.9±11.3	**77.7**±0.1
Ionosphere	70.8±1.8	80.8±2.1	88.6±1.6	67.1±3.9	72.0±1.8	**92.7**±0.6
Arrhythmia	58.2±1.4	60.9±3.3	**61.8**±2.2	22.7±2.5	50.6±3.3	60.4±1.4
Breastw	96.4±0.6	**97.2**±0.5	96.0±0.7	70.6±7.6	63.0±1.8	95.7±0.3
Pima	62.3±1.1	**69.6**±1.2	65.3±1.0	65.9±2.9	55.4±1.7	68.8±0.6
Vowels	4.8±1.0	25.8±4.7	64.4±3.7	23.2±3.2	18.0±4.6	**88.7**±1.6
Letter	12.9±0.7	15.6±3.3	45.0±2.6	14.2±2.3	17.4±2.2	**71.4**±1.9
Cardio	65.0±1.4	73.5±4.1	67.6±0.9	43.0±2.5	43.9±2.7	**78.1**±0.1
Seismic	29.2±1.3	**73.9**±1.5	30.6±1.4	29.2±1.6	24.1±3.2	26.2±0.7
Musk	49.6±1.2	52.0±15.3	**100.0**±0.0	35.4±0.0	38.4±6.5	**100**±0.0
Speech	3.3±0.0	4.9±1.9	5.1±1.0	2.0±1.9	3.9±2.8	**9.3**±0.8
Thyroid	30.8±0.5	**78.9**±2.7	57.3±1.3	72.0±3.2	31.9±4.7	77.0±0.6
Abalone	50.3±6.4	53.4±1.7	43.4±4.8	58.6±1.6	36.9±6.4	**59.7**±0.1
Optdigits	3.0±0.3	15.8±4.3	**90.0**±1.2	22.5±16.8	1.3±0.7	62.0±2.7
Satimage2	77.9±0.9	86.5±1.7	93.8±1.2	35.5±0.4	47.9±3.4	**94.8**±0.8
Satellite	56.7±0.2	69.6±0.5	**76.3**±0.4	46.9±3.7	55.4±1.3	74.6±0.7
Pendigits	34.9±0.6	52.1±6.4	91.0±1.4	44.6±5.3	16.3±2.6	**92.5**±1.3
Annthyroid	31.5±0.5	57.3±1.3	37.8±0.6	**65.4**±2.7	32.1±0.8	57.7±0.6
Mnist	38.5±0.4	51.2±2.5	69.4±0.9	32.6±5.7	33.5±1.7	**71.8**±0.3
Mammo.	**53.4**±0.9	39.0±3.3	38.8±1.5	28.1±4.3	27.1±1.9	43.6±0.5
Shuttle	96.0±0.0	96.4±0.8	97.3±0.2	70.7±1.0	32.0±2.2	**98.2**±0.3
Mullcross	66.0±0.1	99.1±0.5	**100.0**±0.0	67.4±2.1	**100.0**±0.0	**100.0**±0.0
Forest	18.2±0.2	11.1±1.6	**92.1**±0.3	8.1±2.8	9.9±1.5	58.0±10.0
Campaign	**49.5**±0.1	42.4±1.0	41.6±0.4	42.4±0.2	36.6±0.1	**49.8**±0.3
Fraud	44.7±0.9	30.3±3.7	**60.5**±1.5	41.0±0.9	17.1±0.4	**58.1**±3.2
Backdoor	13.4±0.4	3.8±1.2	**88.5**±0.1	3.4±0.2	24.5±0.1	84.1±0.1
mean	44.2	52.6	65.8	39.8	35.6	**68.8**
mean std	**1.5**	3, 9	2.2	3.6	4.0	2.0
mean rank	9.7	7.0	4.9	10.7	11.7	**3.0**

anomalies is equal to the number of existing anomalies. We report the results in tables 1, 2, and 6 in App. C. Most metrics are obtained from [43], apart from NeuTraL-AD [32] which we trained using their official code made available online, and the experiments on the fraud, campaign and backdoor datasets. We evaluate the different methods using both the F1-Score (↑) and AUROC (↑) metrics. We compare our method to both recent deep methods, namely GOAD [4], DROCC [12], NeuTraL-AD [32] and the contrastive approach proposed in [43], and classical non-deep methods such as Isolation Forest [25], KNN [33], RRCF [14], COPOD [24] and PIDForest [10]. We refer the reader to [43] for implementation details of non-deep models. Notice that for DROCC [12], GOAD [4], and NeuTraL-AD [32], we report in table 1 the architecture that obtained the highest mean F1-Score. The metrics obtained for the other architectures are detailed in table 8, 9, and 10 in App. C.1. The mean rank, provided in table 1 and 2, was computed including each architecture of each approach. Following the literature, we report the average metrics over 20 runs. Our model was trained for each dataset on 4 or 8 Nvidia GPUs V100 16Go/32Go depending on the dataset dimension. Note that for small and medium datasets, the model can also be trained on a single GPU.

For each dataset, we considered the same NPT architecture composed of 4 layers alternating between Attention Between Datapoints and Attention Between Attributes and 4 attention heads. Per [23], we consider a Row-wise feed-forward (rFF) network with one hidden layer, 4x expansion factor, GeLU activation, and also include dropout with $p = 0.1$ for both attention weights and hidden layers. We used LAMB [53] with $\beta = (0.9, 0.999)$ as the optimizer and also included a Lookahead [54] wrapper with slow update rate $\alpha = 0.5$ and $k = 6$ steps between updates as in [23]. Similarly, following [23],

we consider a flat-then-anneal learning rate schedule: flat at the base learning rate for 70% of steps and then anneals following a cosine schedule to 0 by the end of the training phase, and set gradient clipping at 1. We chose r in accordance with the masking probability p_{mask} used during training and the total number of features d. We hypothesized that a too-high value of r for a low p_{mask} would pollute the anomaly score with reconstructions too challenging for the model, leading to high reconstruction error for both *normal* samples and anomalies. Moreover, the hardest reconstructions, *i.e.* those with a high number of masked features, would constitute a too high share of the total masks. Indeed, for a fixed d, $\binom{d}{k}$ as a function of k is non-decreasing for $k \leq d/2$ and has an exponential growth rate. Furthermore, raising the value of the parameter r can lead to a substantial augmentation in the number of masks m, consequently inducing a significant upsurge in the inference runtime. We detail in App. B.2 the varying hyperparameters used for each dataset in our experiments. Notice that for most datasets, the hyperparameters remain unchanged. Variations of the hyperparameters are motivated by a swifter convergence of the training loss or computational costs for larger datasets. Each experiment can be replicated using the code made available on github[4].

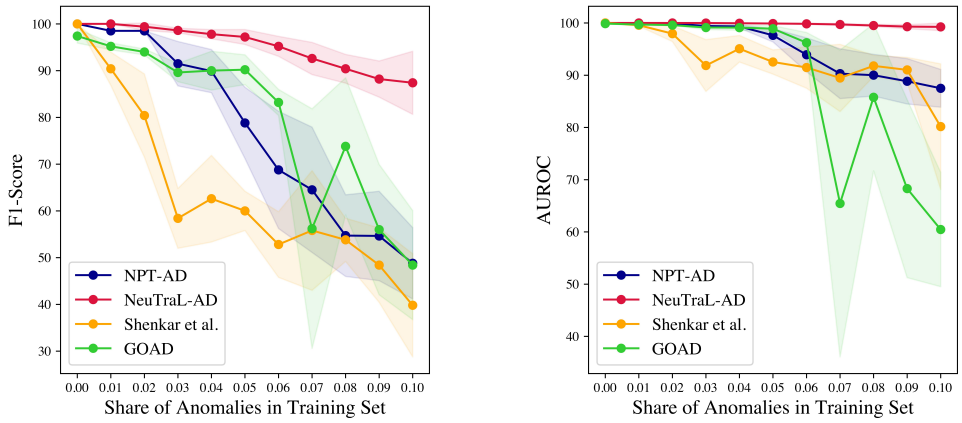
Results As seen in table 1 and 2, our model surpasses existing methods on most datasets by a significant margin regarding the F1-Score. Moreover, our approach displays the highest mean F1-Score and mean rank over all datasets out of the 17 tested approaches. KNN ranks as the second highest in terms of average F1-score and [43] displays the second highest mean rank over all datasets. Also, our approach displays a smaller variance than competing methods except for COPOD, which performs significantly worse than our approach regarding the F1-Score and AUROC. The smaller variance could originate from the fact that our model uses, in a non-parametric fashion, the training set in the inference phase. This contributes to flattening the variations in the anomaly score attributed to discrepancies in the model's weights between runs. We also display in table 6 in App. C.1 the AUROC for the same experiments and observe that we obtain the highest mean AUROC and the lowest mean rank while also displaying a smaller variance than other tested approaches.

5 Discussion

5.1 Training set contamination

Real-life anomaly detection applications often involve contaminated training sets; anomaly detection models must therefore be robust to small levels of dataset contamination. We experimented using a synthetic dataset to evaluate how much NPT-AD suffers from dataset contamination compared to recent deep AD methods. We constructed a synthetic dataset using two perfectly separable distributions for *normal* and anomaly samples. Our training set contained 900 *normal* samples, and we kept aside 100 anomaly samples that we could add to the training set. We considered 11 different training sets with contamination shares ranging from 0% to 10% with a 1% step while keeping the validation set constant with a fixed composition of 10% anomalies and 90% *normal* samples. We display the results of this experiment in Figure 2 in which we show how the performance of NPT-AD varies when the contamination share increases in comparison with NeuTraL-AD [32], GOAD [4] and the internal contrastive approach of [43]. We did not include DROCC in the latter figure since too big error bars caused the graph to be difficult to analyze. We display the figure containing all five approaches, including DROCC, in Figure 3 in App.C.2. Our experimental results show that, as expected, the performance of NPT-AD deteriorates as the proportion of anomalies in the training set increases. For contamination shares lower than 2% (resp. 4%), the F1-Score (resp. AUROC) remains close to its maximum value of 100%. However, the F1-Score and AUROC deteriorate significantly for higher contamination levels while displaying a higher standard deviation. When anomalies constitute 10% of the training set, our approach achieves an average F1-Score slightly lower than 50% and an average AUROC of 87%. We observe that NPT-AD suffers less from dataset contamination than [43] and DROCC [12] for both F1-Score and AUROC. We also notice that DROCC [12] and the approach proposed in [43] are particularly sensible to dataset contamination regarding the F1-Score in comparison with NeuTraL-AD [32], GOAD [4] and NPT-AD even for low contamination shares. Finally, this experiment also highlights that NeuTraL-AD [32] appears significantly more robust than other tested deep methods to training set contamination even for large contamination values.

[4] https://github.com/hugothimonier/NPT-AD/



(a) F1-Score (↑)

(b) AUROC (↑)

Figure 2: Training set contamination impact on the F1-score and AUROC. Each model was trained 5 times for each contamination share. The architecture used for NPT-AD is the same as for all experiments (see section 4). The NPT was trained for 100 epochs with batch size equal to the dataset size, with learning rate 0.01, optimizer LAMB [53] with $\beta = (0.9, 0.999)$, per-feature embedding dimension 16, r set to 1, and masking probability $p_{mask} = 0.15$. NeuTraL-AD [32] and GOAD [4] were trained with hyperparameters as for the thyroid dataset in the original papers and [43] with its default parameters in their implementation.

5.2 Sample-sample dependencies ablation study

Table 3: Ablation study. Variation of the F1-Score and AUROC when preventing NPT from attending to sample-sample interactions. Average difference over 20 runs. All hyperparameters are kept unchanged.

	Mammo.	Glass	BreastW	Pendigits
$\Delta F1$	-1.0	-9.6	-0.5	-2.8
ΔAUROC	-0.1	-0.1	-0.1	-0.1

To investigate the impact of sample-sample dependencies on the effectiveness of our proposed model in detecting anomalies, we conduct an ablation study by shuffling the columns of the unmasked training samples used to reconstruct the test samples. This procedure essentially prevents the Non-Parametric Transformer (NPT) from considering other samples when reconstructing masked features, as elaborated in [23]. Our experiment was carried out on a selected subset of datasets, and is summarized in table 3. Notably, our findings indicate a significant reduction in the F1-score across the tested datasets, whereas the AUROC exhibits a comparatively smaller change. The reduction in F1-score is particularly significant on glass and breastw datasets, emphasizing the role of sample-sample dependencies on these datasets.

To further explore the combined impact of sample-sample and feature-feature dependencies, we introduce a reconstruction-based technique similar to NPT-AD but that relies on KNN imputation for reconstructing masked features (see alg. 1 in appendix C.3). This approach, Mask-KNN, can be seen as approximately equivalent to NPT-AD without considering the feature-feature dependencies. Our experimentation, is detailed in appendix C.3 and summarized in table 11. We observe that Mask-KNN achieves competitive performance on numerous datasets where NPT-AD also performs such as pendigits and speech. However, it notably lags behind on other datasets where NPT-AD performs well, like forestcover and thyroid. Furthermore, NPT-AD consistently outperforms Mask-KNN on most datasets where the method proposed in [43] excels, underscoring the pivotal role of feature-feature dependencies in specific dataset contexts. Additionally, the results presented in tables 3 and 11 align with our observations regarding the datasets glass and breastw: as indicated by the performance of Mask-KNN, on these datasets sample-sample dependencies play a crucial role in

anomaly detection through masking. Overall, the results in tables 11 and 3 highlight the importance of considering both types of dependencies to accurately identify anomalies.

6 Limitations and Conclusion

Limitations As with most non-parametric models, NPT-AD tends to display higher complexity than parametric approaches. NPT-AD can scale well for datasets with a reasonable number of features d; however, for large values of d, our approach involves a high computational cost in terms of memory and time. This cost originates from the complexity of NPT itself and how the anomaly score is derived. In table 13 in appendix C.4 we observe that NPT-AD displays longer runtime for datasets with large values of d when n is also high, *e.g.* Mnist or backdoor. Two factors can account for this, first, the number of reconstruction highly depends on d which increases the inference runtime, secondly due to the feature embeddings, the dimension of the model also increases rapidly with d.

Conclusion In this work, we have proposed a novel deep anomaly detection method designed explicitly for tabular datasets. To the best of our knowledge, our approach is the first to utilize both feature-feature and sample-sample dependencies to identify anomalies. Using an extensive benchmark of tabular datasets, our experiments have demonstrated the effectiveness of our approach, outperforming existing state-of-the-art methods in terms of F1-score and AUROC. Our experiments further demonstrate the robustness of our method to a small training set contamination. This work emphasizes the importance of leveraging sample-sample dependencies to detect anomalies on tabular datasets effectively. Overall, our work invites further exploration of the potential of NPTs for other tasks on tabular data.

Acknowledgment This work was granted access to the HPC resources of IDRIS under the allocation 2023-101424 made by GENCI.
This research publication is supported by the Chair "Artificial intelligence applied to credit card fraud detection and automated trading" led by CentraleSupelec and sponsored by the LUSIS company. The authors would also like to thank Gabriel Kasmi for his helpful advice and feedback and Julien Despois for proofreading the final manuscript.

References

[1] Sercan Ö. Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):6679–6687, May 2021. doi: 10.1609/aaai.v35i8.16826. URL https://ojs.aaai.org/index.php/AAAI/article/view/16826.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. URL https://arxiv.org/abs/1409.0473.

[4] Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1lK_lBtvS.

[5] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, may 2000. ISSN 0163-5808. doi: 10.1145/335191.335388. URL https://doi.org/10.1145/335191.335388.

[6] Xiaoran Chen and Ender Konukoglu. Unsupervised detection of lesions in brain MRI using constrained adversarial auto-encoders. In *Medical Imaging with Deep Learning*, 2018. URL https://openreview.net/forum?id=H1nGLZ2oG.

[7] Penny Chong, Lukas Ruff, Marius Kloft, and Alexander Binder. Simple and effective prevention of mode collapse in deep one-class classification. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, jul 2020. doi: 10.1109/ijcnn48605.2020.9207209. URL https://doi.org/10.11092Fijcnn48605.2020.9207209.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

[9] 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. URL https://openreview.net/forum?id=YicbFdNTTy.

[10] Parikshit Gopalan, Vatsal Sharan, and Udi Wieder. Pidforest: Anomaly detection and certification via partial identification. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:202766416.

[11] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=i_Q1yrOegLY.

[12] Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. Drocc: Deep robust one-class classification. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3711–3721. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/goyal20c.html.

[13] Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=Fp7__phQszn.

[14] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. Robust random cut forest based anomaly detection on streams. In *ICML*, 2016. URL https://www.amazon.science/publications/robust-random-cut-forest-based-anomaly-detection-on-streams.

[15] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. ADBench: Anomaly detection benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=foA_SFQ9zo0.

[16] Sahand Hariri, Matias Carrasco Kind, and Robert J. Brunner. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1479–1489, 2021. doi: 10.1109/TKDE.2019.2947676.

[17] Douglas M. Hawkins. The detection of errors in multivariate data using principal components. *Journal of the American Statistical Association*, 69(346):340–344, 1974. ISSN 01621459. URL http://www.jstor.org/stable/2285654.

[18] Waleed Hilal, S. Andrew Gadsden, and John Yawney. Financial fraud: A review of anomaly detection techniques and recent advances. *Expert Systems with Applications*, 193:116429, 2022. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2021.116429. URL https://www.sciencedirect.com/science/article/pii/S0957417421017164.

[19] Alan Jeffares, Tennison Liu, Jonathan Crabbé, Fergus Imrie, and Mihaela van der Schaar. TANGOS: Regularizing tabular neural networks through gradient orthogonalization and specialization. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=n6H86gW8u0d.

[20] Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. Well-tuned simple nets excel on tabular datasets. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[21] 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 Representations*, 2020.

[22] Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. A mutual information maximization perspective of language representation learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Syx79eBKwr.

[23] Jannik Kossen, Neil Band, Clare Lyle, Aidan Gomez, Tom Rainforth, and Yarin Gal. Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=wRXzOa2z5T.

[24] Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. COPOD: Copula-based outlier detection. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, nov 2020. doi: 10.1109/icdm50108.2020.00135. URL https://doi.org/10.11092Ficdm50108.2020.00135.

[25] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008. doi: 10.1109/ICDM.2008.17.

[26] Philipp Liznerski, Lukas Ruff, Robert A. Vandermeulen, Billy Joe Franks, Marius Kloft, and Klaus Robert Muller. Explainable deep one-class classification. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=A5VV3UyIQz.

[27] Ritesh K. Malaiya, Donghwoon Kwon, Jinoh Kim, Sang C. Suh, Hyunjoo Kim, and Ikkyun Kim. An empirical evaluation of deep learning for network anomaly detection. In *2018 International Conference on Computing, Networking and Communications (ICNC)*, pages 893–898, 2018. doi: 10.1109/ICCNC.2018.8390278.

[28] Mary M. Moya and Don R. Hush. Network constraints and multi-objective optimization for one-class classification. *Neural Netw.*, 9(3):463–474, apr 1996. ISSN 0893-6080. doi: 10.1016/0893-6080(95)00120-4. URL https://doi.org/10.1016/0893-6080(95)00120-4.

[29] Niki J. Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning (ICML)*, 2018. URL http://proceedings.mlr.press/v80/parmar18a.html.

[30] Emanuel Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065 – 1076, 1962. doi: 10.1214/aoms/1177704472. URL https://doi.org/10.1214/aoms/1177704472.

[31] Emanuele Principi, Fabio Vesperini, Stefano Squartini, and Francesco Piazza. Acoustic novelty detection with adversarial autoencoders. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3324–3330, 2017. doi: 10.1109/IJCNN.2017.7966273.

[32] Chen Qiu, Timo Pfrommer, Marius Kloft, Stephan Mandt, and Maja Rudolph. Neural transformation learning for deep anomaly detection beyond images. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8703–8714. PMLR, 2021. URL http://proceedings.mlr.press/v139/qiu21a.html.

[33] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *ACM Sigmod Record*, volume 29, pages 427–438. ACM, 2000.

[34] Tal Reiss and Yedid Hoshen. Mean-shifted contrastive loss for anomaly detection. *arXiv preprint arXiv:2106.03844*, 2021.

[35] Esteban Reyes and Pablo A. Estévez. Transformation based deep anomaly detection in astronomical images. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020. doi: 10.1109/IJCNN48605.2020.9206997.

[36] Stephen Roberts and Lionel Tarassenko. A probabilistic resource allocating network for novelty detection. *Neural Computation*, 6(2):270–284, 1994. doi: 10.1162/neco.1994.6.2.270.

[37] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/ruff18a.html.

[38] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HkgH0TEYwH.

[39] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In Marc Niethammer, Martin Styner, Stephen Aylward, Hongtu Zhu, Ipek Oguz, Pew-Thian Yap, and Dinggang Shen, editors, *Information Processing in Medical Imaging*, pages 146–157, Cham, 2017. Springer International Publishing. ISBN 978-3-319-59050-9.

[40] Thomas Schlegl, Philipp Seeböck, Sebastian Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54, 01 2019. doi: 10.1016/j.media.2019.01.010.

[41] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 582–588, Cambridge, MA, USA, 1999. MIT Press.

[42] Ira Shavitt and Eran Segal. Regularization learning networks: Deep learning for tabular datasets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/500e75a036dc2d7d2fec5da1b71d36cc-Paper.pdf.

[43] Tom Shenkar and Lior Wolf. Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*, 2022.

[44] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. In *8th ICML Workshop on Automated Machine Learning (AutoML)*, 2021. URL https://openreview.net/forum?id=vdgtepS1pV.

[45] João Vitor Valle Silva, Martin Andreoni Lopez, and Diogo M. F. Mattos. Attackers are not stealthy: Statistical analysis of the well-known and infamous kdd network security dataset. In *2020 4th Conference on Cloud and Internet of Things (CIoT)*, pages 1–8, 2020. doi: 10.1109/CIoT50422.2020.9244289.

[46] Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. Learning and evaluating representations for deep one-class classification. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=HCSgyPUfeDj.

[47] Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C. Bayan Bruss, and Tom Goldstein. SAINT: improved neural networks for tabular data via row attention and contrastive pre-training. *CoRR*, abs/2106.01342, 2021. URL https://arxiv.org/abs/2106.01342.

[48] David Tax and Robert Duin. Support vector data description. *Machine Learning*, 54:45–66, 01 2004. doi: 10.1023/B:MACH.0000008084.60811.49.

[49] Hugo Thimonier, Fabrice Popineau, Arpad Rimmel, Bich-Liên Doan, and Fabrice Daniel. TracInAD: Measuring influence for anomaly detection. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, 2022. doi: 10.1109/IJCNN55064.2022.9892058. URL https://doi.org/10.1109/IJCNN55064.2022.9892058.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[51] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, Jie Chen, Zhaogang Wang, and Honglin Qiao. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 187–196, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN 9781450356398. doi: 10.1145/3178876.3185996. URL https://doi.org/10.1145/3178876.3185996.

[52] Sun Yanmin, Andrew Wong, and Mohamed S. Kamel. Classification of imbalanced data: a review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23:687–719, 11 2011. doi: 10.1142/S0218001409007326.

[53] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Syx4wnEtvH.

[54] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/90fd4f88f588ae64038134f1eeaa023f-Paper.pdf.

[55] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

Appendix A Multi-Head Self-Attention

Scaled dot-product attention as first proposed in [50] describes a mapping between queries $Q_i \in \mathbb{R}^{1\times h_k}$, keys $K_i \in \mathbb{R}^{1\times h_k}$ and values $V_i \in \mathbb{R}^{1\times h_v}$ to an output. The output is computed as a weighted sum of the values, where each weight is obtained by measuring the compatibility between queries and keys. Take $\mathbf{Q} \in \mathbb{R}^{n\times h_k}$, $\mathbf{K} \in \mathbb{R}^{m\times h_k}$ and $\mathbf{V} \in \mathbb{R}^{m\times h_v}$ the corresponding matrices in which queries, keys, and values are stacked. Scaled dot-product attention is computed as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{h_k}}\right)\mathbf{V} \tag{7}$$

where, for convenience, one often sets $h_k = h_v = h$.

To foster the ability of a model to produce diverse and powerful representations of data samples, one often includes several dot-product attention mechanisms. Multi-head dot-product attention then describes the concatenation of k independent attention heads:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \underset{\text{axis=k}}{\text{concat}}(O_1, \dots, O_k)W^O, \text{ where} \tag{8}$$

$$O_j = \text{Attention}(\mathbf{Q}W_j^Q, \mathbf{K}W_j^K, \mathbf{V}W_j^V) \tag{9}$$

where the embedding matrices $W_j^Q, W_j^K, W_j^V \in \mathbb{R}^{h\times h/k}$ are learned for each attention head $j \in \{1, \dots, k\}$ and $W^O \in \mathbb{R}^{h\times h}$ serves to mix the h attention heads outputs. NPTs only include multi-head *self*-attention mechanisms which consist in multi-head dot-product attention where queries, keys, and values are identical:

$$\text{MHSelfAtt}(\mathbf{H}) = \text{MultiHead}(\mathbf{Q} = \mathbf{H}, \mathbf{K} = \mathbf{H}, \mathbf{V} = \mathbf{H}) \tag{10}$$

As described in [23], NPT follows transformer best practices to improve performances and involves a residual branch as well as layer normalization (LN) [2] before MHSelfAtt$(.)$.

$$\text{Res}(\mathbf{H}) = \mathbf{H}W^{\text{res}} + \text{MHSelfAtt}\left(\text{LN}(\mathbf{H})\right) \tag{11}$$

where $W^{\text{res}} \in \mathbb{R}^{h\times h}$ are learned weights. Layer normalization is also added after the residual branch as well as a row-wised feed-forward network (rFF):

$$\text{MHSA}(\mathbf{H}) = \text{Res}(\mathbf{H}) + \text{rFF}\left(\text{LN}\left(\text{Res}\left(\mathbf{H}\right)\right)\right) \in \mathbb{R}^{n\times h} \tag{12}$$

Appendix B Datasets characteristics and experimental settings

B.1 Dataset characteristics

In table 4, we display the main characteristics of the datasets involved in our experiments.

Table 4: Datasets characteristics

Dataset	n	d	Outliers
Wine	129	13	10 (7.7%)
Lympho	148	18	6 (4.1%)
Glass	214	9	9 (4.2%)
Vertebral	240	6	30 (12.5%)
WBC	278	30	21 (5.6%)
Ecoli	336	7	9 (2.6%)
Ionosphere	351	33	126 (36%)
Arrhythmia	452	274	66 (15%)
BreastW	683	9	239 (35%)
Pima	768	8	268 (35%)
Vowels	1456	12	50 (3.4%)
Letter Recognition	1600	32	100 (6.25%)
Cardio	1831	21	176 (9.6%)
Seismic	2584	11	170 (6.5%)
Musk	3062	166	97 (3.2%)
Speech	3686	400	61 (1.65%)
Thyroid	3772	6	93 (2.5%)
Abalone	4177	9	29 (0.69%)
Optdigits	5216	64	150 (3%)
Satimage-2	5803	36	71 (1.2%)
Satellite	6435	36	2036 (32%)
Pendigits	6870	16	156 (2.27%)
Annthyroid	7200	6	534 (7.42%)
Mnist	7603	100	700 (9.2%)
Mammography	11183	6	260 (2.32%)
Shuttle	49097	9	3511 (7%)
Mulcross	262144	4	26214 (10%)
ForestCover	286048	10	2747 (0.9%)
Campaign	41188	62	4640 (11.3%)
Fraud	284807	29	492 (0.17%)
Backdoor	95329	196	2329 (2.44%)

B.2 Experimental settings

Table 5: Datasets hyperparameters. When the batch size is -1 it refers to a full pass over the training set before an update of the weights.

Dataset	epoch	batch size	lr	p_{mask}^{train}	r	m	e
Wine	1000	-1	0.001	0.15	1	13	8
Lympho	100	-1	0.01	0.15	4	3078	16
Glass	1000	-1	0.01	0.15	4	255	16
Vertebral	2000	-1	0.001	0.15	1	6	8
WBC	100	-1	0.01	0.15	3	4525	16
Ecoli	100	-1	0.01	0.15	3	63	16
Ionosphere	100	-1	0.001	0.15	2	561	16
Arrhythmia	100	-1	0.01	0.15	1	274	16
BreastW	500	-1	0.01	0.15	3	129	16
Pima	500	-1	0.01	0.15	4	162	16
Vowels	1000	-1	0.01	0.15	2	78	16
Letter Recognition	1000	-1	0.01	0.15	1	32	16
Cardio	100	-1	0.01	0.15	2	231	16
Seismic	100	-1	0.01	0.15	2	276	16
Musk	100	-1	0.01	0.15	2	166	16
Speech	1000	512	0.001	0.15	1	400	8
Thyroid	5000	-1	0.01	0.1	2	21	16
Abalone	1000	-1	0.0001	0.15	4	162	16
Optdigits	500	-1	0.01	0.2	1	64	16
Satimage-2	100	-1	0.01	0.2	1	36	16
Satellite	100	-1	0.01	0.2	1	36	16
Pendigits	1000	-1	0.01	0.25	2	136	16
Annthyroid	400	-1	0.01	0.15	1	6	16
Mnist	1000	-1	0.001	0.15	1	100	32
Mammography	200	-1	0.01	0.25	4	56	16
Shuttle	100	4096	0.01	0.25	3	129	64
Mulcross	100	4096	0.001	0.15	2	10	16
ForestCover	100	4096	0.01	0.15	2	55	16
Campaign	100	4096	0.001	0.15	1	62	16
Fraud	100	4096	0.001	0.2	1	29	32
Backdoor	1000	4096	0.001	0.2	1	196	32

Appendix C Additional experiments

C.1 Additional results

In this section, we display the metrics for each of the experiments we performed. This includes the AUROC for the approaches for which it is relevant to compute it; displayed in tables 6 and 7, the F1-score for each architecture discussed in the original papers of NeuTraL-AD [32] in table 10, GOAD [4] in table 9 and DROCC [12] in table 8. For each of these tables, we highlight in bold the highest metric in the table.

Table 6: Anomaly detection AUROC(↑). We perform 5% T-test to test whether the differences between the highest metrics for each dataset are statistically significant.

Method	DROCC (Thyroid)	DROCC (Arrhyth.)	DROCC (Abalone)	GOAD (Thyroid)	GOAD (kddrev)	GOAD (kdd)	Internal Cont.	NPT-AD (Ours)
Wine	53.5±22	60.1±32	90.9±8.2	95.2±1.9	97.3±1.7	86.3±9.5	**99.5**±0.6	96.6±0.5
Lympho	6.4±5.2	58.6±30	83.7±12	94.8±5.6	79.7±11	59.9±15	99.5±0.3	**99.9**±0.1
Glass	63.5±9.1	55.5±22	75.4±8.9	62.2±14	85.5±7	82.1±6.3	**88.1**±5.0	82.8±2.4
Vertebral	**55.0**±5.1	**58.0**±15	41.2±10	47.0±13	52.2±3.9	49.4±4.2	51.1±3.2	**54.6**±3.9
WBC	6.8±1.8	41.3±25	35.4±13	95.4±0.7	66.1±12	86.6±2.9	95.4±1.1	**96.3**±0.3
Ecoli	N/A	N/A	N/A	82.7±8.4	87.2±3.3	84.7±6.8	86.5±1.2	**88.7**±1.6
Ionosph.	19.6±5.8	83.5±5.6	80.0±2.8	92.4±1.3	96.3±1.1	96.5±1.1	**98.1**±0.4	97.4±1.7
Arrhyth.	53.2±7.0	52.7±8.6	51.2±8.1	80.0±1.9	73.3±5.1	64.3±8.8	**81.7**±0.6	80.1±0.0
Breastw	7.7±8.6	64.4±33	96.6±3.3	98.7±0.8	80.8±9.5	97.7±0.8	**99.1**±0.3	98.6±0.1
Pima	36.2±4.6	54.9±11	69.1±4.9	68.7±3.9	59.3±2.2	63.2±2.3	59.4±2.8	**73.4**±0.4
Vowels	79.4±9.5	72.0±12	95.3±2.1	81.0±2.4	98.5±0.3	97.6±0.5	**99.7**±0.1	99.3±0.1
Letter	77.6±3.3	73.3±5.4	90.0±1.2	60.9±0.7	89.9±0.5	87.6±0.9	92.8±0.9	**96.1**±0.2
Cardio	84.3±4.0	73.8±12	73.5±3.2	**94.8**±1.7	81.3±4.5	84.6±3.0	92.7±0.8	**94.7**±0.2
Seismic	58.2±2.8	60.3±4.5	56.7±1.3	**69.5**±1.5	67.2±1.2	67.9±1.2	62.9±1.0	**69.8**±0.3
Musk	2.3±5.1	**100**±0.0	**100**±0.0	**100**±0.0	**100**±0.0	**100**±0.0	**100**±0.0	**100**±0.0
Speech	51.2±5.6	50.5±4.0	52.6±3.4	47.1±1.3	**65.3**±3.2	54.1±4.4	58.9±2.7	54.3±0.3
Thyroid	95.6±0.9	96.1±2.5	98.1±0.3	94.5±1.5	77.1±8.8	89.2±3.0	**98.5**±0.1	97.8±0.1
Abalone	82.4±14	52.9±26	70.6±9.7	89.2±0.9	46.0±3.7	54.3±7.8	**94.3**±0.6	91.6±1.2
Optdig.	84.2±4.6	89.0±4.6	89.5±2.1	66.9±3.3	95.7±0.5	93.1±1.9	**97.5**±1.5	**97.5**±0.3
Satimage	19.1±1.4	87.5±8.8	11.5±1.2	99.1±0.1	86.5±7.1	93.2±1.7	99.8±0.1	**99.9**±0.0
Satellite	64.6±8.9	73.1±1.3	50.2±2.2	69.1±0.8	76.3±1.0	78.2±0.9	**80.6**±1.7	**80.3**±0.9
Pendigits	58.9±7.6	50.8±15	76.6±5.4	87.5±3.9	89.2±2.9	85.1±3.4	99.5±0.1	**99.9**±0.0
Annthyr.	92.9±2.3	86.5±3.6	**93.4**±1.3	76.1±6.5	89.6±4.9	93.2±0.9	80.5±1.3	86.7±0.6
Mnist	N/A	N/A	N/A	90.9±0.4	89.4±0.7	87.7±1.0	**98.2**±0.0	94.4±0.1
Mammo.	81.0±1.3	85.0±2.1	82.0±1.5	66.3±6.4	57.2±1.9	54.5±2.3	81.1±2.0	88.6±0.3
Mullcross	N/A	N/A	N/A	100±0.0	N/A	51.3±16	**100**±0.0	**100**±0.0
Shuttle	N/A	N/A	N/A	88.4±5.5	N/A	99.9±0.0	**100**±0.0	99.8±0.1
Forest	N/A	N/A	N/A	15.9±6.6	N/A	76.0±5.3	**96.2**±0.6	**95.8**±7.9
Campaign	N/A	N/A	N/A	38.4±2.0	N/A	50.9±2.5	73.7±1.5	**79.1**±0.5
Fraud	N/A	N/A	N/A	83.5±2.7	N/A	86.3±0.8	95.2±0.5	**95.7**±0.1
Backdoor	N/A	N/A	N/A	61.3±10.2	N/A	88.9±1.1	92.6±0.6	95.2±0.1
mean	39.8.1	50.9	53.7	77.3	64.1	78.8	88.8	**89.8**
mean std	4.5	9.1	3.4	3.5	3.2	3.8	**1.0**	**0.8**
mean rank	9.5	9.2	8	6.8	7.1	6.5	2.7	**2.3**

Table 7: Anomaly detection AUROC(↑). We perform 5% T-test to test whether the differences between the highest metrics for each dataset are statistically significant.

Method	NeuTraL-AD (thyroid)	NeuTraL-AD (arrhythmia)	NeuTraL-AD (kddrev)	NeuTraL-AD (kdd)	COPOD
Wine	82.9±13.1	**95.4**±1.9	86.0±10.3	85.2±12.9	87.5±1.7
Lympho	90.7±5.1	80.9±7.5	93.1±4.1	83.1±9.9	**99.4**±0.4
Glass	62.2±3.0	62.9±1.2	62.5±4.6	**65.1**±2.9	63.7±3.3
Vertebral	32.8±3.5	25.8±4.5	51.9±14.9	**54.4**±16.3	32.6±1.2
Wbc	80.4±5.0	92.6±2.2	62.4±8.2	32.5±11.6	**96.3**±0.5
Ecoli	43.2±9.3	53.8±9.9	52.5±10.6	49.9±10.9	**81.0**±1.2
Ionosphere	87.9±2.9	**95.7**±1.7	92.9±0.9	87.2±2.7	80.3±2.1
Arrhythmia	76.0±1.5	80.2±1.6	77.9±1.8	76.4±2.7	**80.5**±1.3
Breastw	86.0±2.2	96.2±1.1	95.9±1.6	91.3±5.1	**99.4**±0.2
Pima	55.1±1.9	60.6±1.2	57.3±2.1	57.3±3.7	**65.2**±0.7
Vowels	**72.3**±3.1	69.7±3.8	62.2±6.1	56.1±8.0	49.6±1.0
Letter	40.4±3.9	35.4±0.8	36.1±2.9	37.4±4.2	**50.1**±0.8
Cardio	74.6±2.7	74.3±3.1	47.7±4.2	28.9±6.3	**92.2**±0.3
Seismic	42.7±4.2	39.9±5.5	40.6±9.4	41.0±13.1	**70.8**±0.4
Musk	**99.5**±0.2	99.4±0.2	98.2±1.0	91.9±3.6	94.5±0.2
Speech	46.7±1.6	48.0±1.6	52.7±3.2	54.3±2.3	49.1±0.5
Thyroid	**97.5**±0.2	97.1±0.2	95.8±1.6	79.4±10.4	94.1±0.2
Abalone	**92.9**±1.0	92.7±0.8	80.8±2.4	73.9±4.6	86.3±0.3
Optdigits	72.3±7.1	82.6±5.4	**84.1**±4.3	78.4±7.2	68.0±0.4
Satimage	99.6±0.4	**99.8**±0.1	**99.8**±0.1	**99.7**±0.1	97.4±0.1
Satellite	80.7±0.4	**81.0**±0.4	76.8±0.6	74.0±1.6	63.5±0.2
Pendigits	88.6±5.7	**98.6**±0.8	97.5±0.9	93.0±3.1	90.4±0.2
Annthyroid	**87.7**±4.2	80.2±2.9	64.9±1.6	63.7±3.0	77.4±0.4
Mnist	**97.5**±0.3	**97.8**±0.2	93.4±0.7	89.9±1.1	77.2±0.2
Mammography	67.6±3.3	73.8±2.5	65.4±3.3	69.6±3.3	**90.5**±0.1
Mullcross	98.5±2.1	**99.5**±1.5	**99.6**±0.5	99.2±1.5	93.2±0.0
Shuttle	99.8±0.3	**99.9**±0.1	**100.0**±0.0	**99.9**±0.1	99.4±0.0
Forestcover	**96.8**±1.5	**96.7**±1.1	84.4±7.0	82.6±6.2	88.4±0.0
Campaign	75.4±4.8	70.0±2.1	76.5±0.7	76.4±0.5	**78.3**±0.1
Fraud	81.6±7.4	84.8±4.7	93.1±0.6	93.3±0.4	**94.7**±0.1
Backdoor	91.6±5.8	92.5±7.5	**92.9**±0.4	**92.9**±0.3	78.9±0.1
mean	77.5	**79.3**	76.6	72.8	**79.7**
mean std	3.5	2.5	3.6	5.1	**0.6**
mean rank	7.5	**6.5**	7.2	8	6.8

Table 8: DROCC [12]: anomaly detection F1-score (↑) between architecture. The mean rank was computed including all architectures of all models.

Method	DROCC (thyroid)	DROCC (arrhyth.)	DROCC (abalone)
Wine	20.0±19.0	32.0±35.4	**63.0**±20.0
Lympho	0.0±0.0	38.3±23.6	**65.0**±5.0
Glass	**22.2**±17.2	13.3±12.0	14.5±11.1
Vertebral	25.7±5.4	**27.0**±15.9	9.3±6.1
Wbc	0.0±0.0	**18.6**±16.0	9.0±6.2
Ecoli	N/A	N/A	N/A
Ionosph.	29.9±6.8	76.3±6.4	**76.9**±2.8
Arrhyth.	**38.8**±6.2	37.9±8.0	37.1±6.8
Breastw	15.3±7.7	63.8±29.3	**93.0**±3.7
Pima	40.6±3.3	55.2±8.0	**66.0**±4.1
Vowels	33.0±16.4	20.4±15.0	**66.2**±8.8
Letter	39.0±4.8	31.3±6.5	**55.6**±3.6
Cardio	**62.6**±6.1	53.3±12.9	49.8±3.2
Seismic	17.7±2.5	17.9±2.7	**19.1**±0.9
Musk	1.3±3.3	**99.7**±0.9	99.4±1.5
Speech	3.4±2.4	2.1±1.9	**4.3**±2.0
Thyroid	68.4±3.2	69.7±5.7	**72.7**±3.1
Abalone	**44.3**±17.6	11.6±10.5	17.9±1.3
Optdigits	18.4±5.4	26.5±12.6	**30.5**±5.2
Satimage2	10.2±2.5	**33.7**±19.6	4.8±1.6
Satellite	61.3±6.3	**68.1**±0.7	52.2±1.5
Pendigits	7.9±2.9	10.6±7.9	**11.0**±2.6
Annthyr.	63.8±4.7	55.6±5.2	**64.2**±3.3
Mnist	N/A	N/A	N/A
Mammo.	**34.1**±2.2	31.5±6.2	32.6±2.1
Shuttle	N/A	N/A	N/A
Mullcross	N/A	N/A	N/A
Forest	N/A	N/A	N/A
Campaign	N/A	N/A	N/A
Fraud	N/A	N/A	N/A
Backdoor	N/A	N/A	N/A
mean	21.2	28.9	**32.7**
mean std	4.7	8.5	3.4
mean rank	12.6	11.9	10.8

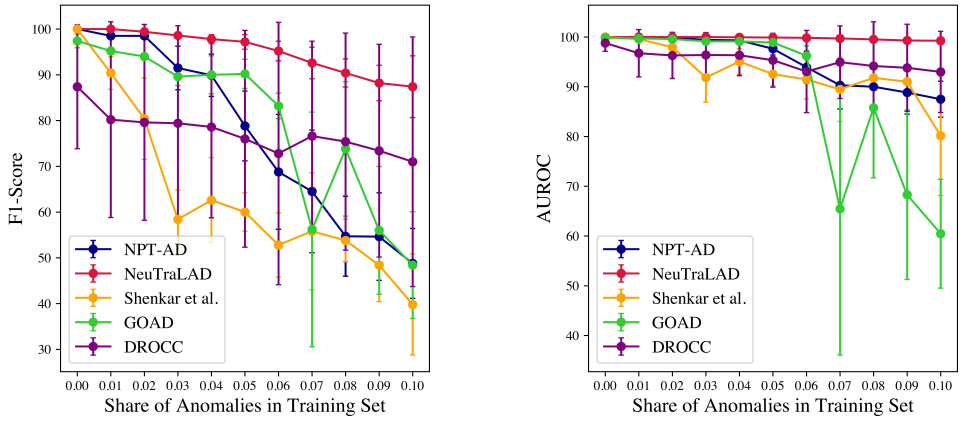
Table 9: GOAD [4]: anomaly detection F1-score (↑) between architecture. The mean rank was computed including all architectures of all models.

Method	GOAD (thyroid)	GOAD (kddrev)	GOAD (kdd)
Wine	67.0±9.4	**76.0**±10.8	42.2±26.9
Lympho	**68.3**±13.0	67.7±7.8	46.0±21.5
Glass	12.7±3.9	**25.7**±12	24.0±15.1
Vertebral	16.3±9.6	**26.9**±5.2	25.5±4.7
Wbc	**66.2**±2.9	16.8±16.1	57.2±6.9
Ecoli	61.4±31.7	**69.3**±23.7	66.1±27.8
Ionosph.	83.4±2.6	88.1±2.3	**88.7**±2.7
Arrhyth.	**52.0**±2.3	51.6±4.0	45.2±7.6
Breastw	**96.0**±0.6	73.5±9.4	94.8±1.0
Pima	**66.0**±3.1	57.3±1.9	60.2±2.0
Vowels	31.1±4.2	**78.6**±2.9	72.6±4.5
Letter	20.7±1.7	**53.8**±2.2	48.6±3.0
Cardio	**78.6**±2.5	48.9±5.8	58.4±4.8
Seismic	**24.1**±1.0	18.6±1.9	19.4±2.6
Musk	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0
Speech	4.8±2.3	**8.9**±2.9	4.4±2.4
Thyroid	**72.5**±2.8	17.2±9.4	32.9±9.9
Abalone	**57.6**±2.2	6.2±1.4	6.6±1.0
Optdigits	0.3±0.3	**45.8**±2.6	36.5±9.9
Satimage2	**90.7**±0.7	20.4±10.5	21.7±2.2
Satellite	64.2±0.4	67.9±2.0	**70.1**±0.8
Pendigits	**40.1**±5.0	25.1±3.6	19.4±4.5
Annthyr.	50.3±6.3	61.4±7.8	**68.0**±3.7
Mnist	66.9±1.3	**67.5**±1.2	66.2±1.5
Mammo.	**33.7**±6.1	16.5±1.3	16.0±1.5
Shuttle	73.5±5.1	N/A	**98.4**±0.2
Mullcross	**99.7**±0.8	N/A	36.4±17.0
Forest	0.1±0.2	N/A	**15.0**±4.3
Campaign	16.2±1.8	N/A	**25.6**±2.1
Fraud	**53.1**±10.2	N/A	**53.7**±2.0
Backdoor	12.7±2.9	N/A	**39.9**±3.2
mean	**50.9**	38.3	47.1
mean std	4.4	4.8	6.4
mean rank	7.8	9.5	8.4

Table 10: NeuTraL-AD [32]: anomaly detection F1-score (↑) between architecture. The mean rank was computed including all architectures of all models.

Method	NeuTraL-AD (thyroid)	NeuTraL-AD (arrhythmia)	NeuTraL-AD (kddrev)	NeuTraL-AD (kdd)
Wine	51.4±26.2	**78.2**±4.5	62.3±26.9	62.3±28.9
Lympho	46.7±17.9	20.0±18.7	**54.2**±15.7	34.2±15.3
Glass	7.5±6.2	9.0±4.4	9.5±5.9	**13.0**±7.8
Vertebral	9.2±3.4	3.8±1.2	**23.3**±9.8	13.0±7.8
Wbc	40.7±10.0	**60.9**±5.6	21.1±10.0	13.0±7.8
Ecoli	4.0±5.8	7.0±7.1	6.5±11.1	**8.0**±12.5
Ionosph.	79.2±2.8	**90.6**±2.4	86.9±1.4	**79.4**±4.0
Arrhyth.	54.9±3.4	**59.5**±2.6	57.7±1.6	57.2±3.1
Breastw	80.2±2.0	**91.8**±1.3	89.6±2.9	85.6±5.6
Pima	55.4±1.7	**60.3**±1.4	57.2±1.9	56.8±2.5
Vowels	**13.2**±6.3	10.0±6.2	5.0±3.8	3.9±3.4
Letter	4.9±1.7	**5.7**±0.8	3.6±1.2	4.8±2.8
Cardio	**46.9**±3.9	45.5±4.3	14.7±5.0	3.8±2.7
Seismic	**12.8**±1.3	11.8±4.3	8.7±4.4	10.7±7.9
Musk	98.9±0.0	**99.0**±0.0	79.3±11.2	43.4±16.5
Speech	5.3±1.8	4.7±1.4	4.3±2.4	**6.1**±2.7
Thyroid	**75.6**±2.3	69.4±1.4	61.4±8.4	26.6±17.0
Abalone	**60.8**±4.2	53.2±4.0	45.6±8.6	47.1±8.3
Optdigits	11.9±7.0	16.2±7.3	**18.5**±9.6	17.1±9.4
Satimage2	85.8±9.0	92.3±1.9	**92.4**±1.4	91.3±0.9
Satellite	**72.7**±0.3	71.6±0.6	70.4±0.6	66.9±2.1
Pendigits	32.4±14.3	**69.8**±8.7	58.4±8.9	42.2±13.2
Annthyr.	**53.5**±5.1	44.1±2.3	33.2±2.1	29.1±4.3
Mnist	82.8±0.9	**84.8**±0.5	68.8±2.5	60.8±2.8
Mammo.	11.3±1.7	19.2±2.4	**20.8**±3.7	19.1±4.9
Shuttle	97.1±0.4	**97.9**±0.2	97.6±0.1	97.5±0.1
Mullcross	88.9±12.2	**96.3**±10.5	96.2±3.6	92.9±9.4
Forest	**64.6**±9.9	51.6±8.2	12.2±11.4	8.7±7.6
Campaign	**52.0**±4.1	42.1±1.7	51.6±0.1	51.2±0.8
Fraud	24.7±7.8	24.3±7.8	**61.0**±5.2	55.9±4.2
Backdoor	84.9±5.0	84.4±1.8	**87.3**±0.2	86.8±0.3
mean	48.7	**50.8**	47.0	41.7
mean std	5, 8	4.0	5.9	7.0
mean rank	9.8	9.0	9.4	10.7

C.2 Contamination



(a) F1-score (↑) (b) AUROC (↑)

Figure 3: Training set contamination impact on the F1-score and AUROC.

C.3 Mask-KNN

To further investigate the impact of combining feature-feature and sample-sample dependencies, we rely on reconstruction-based strategy which makes use of the KNN-Imputer strategy.

K-Nearest Neighbor Imputation Take a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ and for which some samples might display missing values in the feature vector. K-nearest neighbor imputation for a sample $\mathbf{z} \in \mathcal{D}$ consists in identifying the k nearest neighbors of sample $\mathbf{z}$ given a distance measure $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, where k is a hyperparameter that must be discretionary chosen. This distance measure only takes into account the non-missing features of sample $\mathbf{z}$. Let $\mathcal{I}$ designate the index of the non-missing features and $\mathbf{z}^{[\mathcal{I}]}$ the corresponding features of sample $\mathbf{z}$, then the k-nearest neighbors of sample $\mathbf{z}$ are identified through evaluating the distance $d(\mathbf{z}^{[\mathcal{I}]}, \mathbf{x}_j^{[\mathcal{I}]})$ for each $\mathbf{x}_j \in \mathcal{D}$ and ordering them to find the k smallest. Let $\mathcal{K}(z)$ designate the k nearest neighbors of sample $\mathbf{z}$, $\bar{\mathcal{I}}$ the missing values of $\mathbf{z}$, then $\forall i \in \bar{\mathcal{I}}$

$$\hat{z}^i = \frac{1}{k} \sum_{\mathbf{x} \in \mathcal{K}(\mathbf{z})} \mathbf{x}^i. \tag{13}$$

Other imputation methods include weighting each sample in $\mathcal{K}(\mathbf{z})$ by its inverse distance to $\mathbf{z}$, denoted $\omega_{(\mathbf{z},\mathbf{x})}^{[\mathcal{I}]} = 1/d(\mathbf{z}^{[I]}, \mathbf{x}_j^{[\mathcal{I}]})$. This gives

$$\hat{\mathbf{z}}^i = \frac{1}{\sum_{\mathbf{x}} \omega_{(\mathbf{z},\mathbf{x})}^{[\mathcal{I}]}} \sum_{\mathbf{x} \in \mathcal{K}(\mathbf{z})} \omega_{(\mathbf{z},\mathbf{x})}^{[\mathcal{I}]} \mathbf{x}^i. \tag{14}$$

Mask-KNN Anomaly Score Consider a training set $\mathcal{D}_{train} = \{\mathbf{x}_i\}_{i=1}^{n_{train}}$, $\mathbf{x}_i \in \mathbb{R}^d$ comprised of only *normal* samples and a validation set $\mathcal{D}_{val} = \{\mathbf{x}_i\}_{i=1}^{n_{val}}$ for which we wish to predict the label. In a reconstruction-based approach we construct an anomaly score based on how masked samples are well-reconstructed using KNN imputation as described in the previous paragraph. First, we construct a mask bank comprised of m masks, where $m = \sum_{j=1}^{r} \binom{d}{j}$ and r designates the maximum number of features masked simultaneously. The mask bank is comprised of all possible combinations of j masked features for $j \leq r$. Each mask corresponds to a d-dimensional vector composed of 0 and 1, where 1's indicate that the corresponding features will be masked. Let us denote as $\hat{\mathbf{z}}^{(\ell)}$ the reconstructed sample $\mathbf{z}$ for mask ℓ, take $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a distance measure, *e.g.* the ℓ_2-norm, then

the anomaly score for sample $\mathbf{z}$ is given as

$$\text{Mask-KNN}(\mathbf{z}) = \sum_{\ell=1}^{m} d(\mathbf{z}, \hat{\mathbf{z}}^{(\ell)}) \tag{15}$$

We give the pseudo-code of this method in alg. 1.

Algorithm 1 Pseudo Python Code for Mask-KNN

Require: $\mathcal{D}_{train} \in \mathbb{R}^{n_{train} \times d}$, $\mathcal{D}_{val} \in \mathbb{R}^{n_{val} \times d}$, k, $mask\_bank$, $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$
  Mask-KNN $\leftarrow dict()$
  $B \leftarrow$ random sample of size b from $\mathcal{D}_{train}$
  for `mask` $\in mask\_bank$ **do**
    for `idx` $\in$ `range`(n_{val}) **do**
      $\mathbf{z} \leftarrow \mathcal{D}_{val}[\texttt{idx}, :]$
      $\tilde{\mathbf{z}} \leftarrow apply\_mask(z, \texttt{mask})$
      $\mathbf{X} \leftarrow (\tilde{\mathbf{z}}, B)^\top$
      $\hat{\mathbf{X}} \leftarrow$ KNNImputer$(\mathbf{X}, k)$
      $\hat{\mathbf{z}} \leftarrow \hat{\mathbf{X}}[0, :]$
      Mask-KNN$[\texttt{idx}] \mathrel{+}= d(\mathbf{z}, \hat{\mathbf{z}})$
    end for
  end for

Implementation For simplicity we set r to 2 for all experiments, except for large dataset ($n > 200,000$) for which r was set to 1 for computational reasons. We set k, the number of neighbors, to 5 as for the vanilla KNN implementation. When present, categorical features were encoded using one-hot encoding. Except for large datasets ($n > 200,000$) with many features, d, such as ForestCover, Fraud and Backdoor, we set B as the entire training set. Otherwise, we take a random subsample of size $b = 10,000$. We use the imputation strategy described in equation 14 to reconstruct the masked sampled. We report the results of this experiment in table 11 and compare the performance of Mask-KNN to KNN, the internal contrastive approach of [43] and NPT-AD. We run the algorithm 20 times for each dataset, except for ForestCover, Fraud and Backdoor, for which report an average over 10 runs for computational reasons. The mean rank, provided in table 11, was computed, including each architecture of each approach. For completeness, we also include a table containing the mean rank of all approaches including MasK-KNN in table 12.

Results We observe that Mask-KNN obtains satisfactory results on a significant share of the tested datasets, *e.g.* pendigits, satellite; while also displaying poor performance on some datasets such as forest or backdoor in comparison with NPT-AD. Several factors can account for this. First, NPTs automatically select the number of relevant samples on which to rely to reconstruct the masked features, thus making this approach much more flexible than Mask-KNN, which has a fixed number of neighbors. Second, NPT-AD relies on attention mechanisms to learn the weights attributed to relevant samples while Mask-KNN relies on the ℓ_2-distance. Although the ℓ_2-distance offers a precise measure of similarity based on geometric distance, the attention mechanism can capture much more complex relations between samples. Finally, NPT-AD not only relies on sample-sample dependencies to reconstruct the mask features, but it also attends to feature-feature dependencies.

The strong performance of NPT-AD on datasets where Mask-KNN also performs well serves as evidence supporting the fact that NPT-AD effectively captures sample-sample dependencies. Moreover, NPT-AD outperforms Mask-KNN on most datasets where the approach of [43] performs well, highlighting the crucial role of feature-feature dependencies on specific datasets. The results displayed in table 11 show that NPT-AD manages to capture both feature-feature and sample-sample dependencies to reconstruct samples when sample-sample dependencies are not sufficient.

Table 11: Anomaly detection F1-score (↑). We perform 5% T-test to test whether the difference between the highest metrics for each dataset is statistically significant. **Apart from this table, Mask-KNN was not included in the computation of the mean rank.** The mean rank for the F1-score of all approaches including Mask-KNN is displayed in table 12.

Method	Internal Cont.	KNN	NPT-AD	Mask-KNN
Wine	90.0±6.3	**94.0**±4.9	72.5±7.7	28.0±18.1
Lympho	86.7±6.0	80.0±11.7	**94.2**±7.9	60.0±12.2
Glass	**27.2**±10.6	11.1±9.7	**26.2**±10.9	**26.7**±5.4
Vertebral	**26.0**±7.7	10.0±4.5	20.3±4.8	**24.7**±5.9
Wbc	**67.6**±3.6	63.8±2.3	**67.3**±1.7	**68.1**±3.0
Ecoli	70.0±7.8	**77.8**±3.3	**77.7**±0.1	63.9±6.9
Ionosph.	**93.2**±1.3	88.6±1.6	**92.7**±0.6	89.7±0.9
Arrhyth.	61.8±1.8	61.8±2.2	60.4±1.4	**62.9**±2.4
Breastw	**96.1**±0.7	**96.0**±0.7	**95.7**±0.3	**96.2**±0.7
Pima	59.1±2.2	65.3±1.0	**68.8**±0.6	63.5±1.8
Vowels	**90.8**±1.6	64.4±3.7	88.7±1.6	84.3±4.9
Letter	62.8±2.4	45.0±2.6	**71.4**±1.9	56.7±3.2
Cardio	71.0±2.4	67.6±0.9	**78.1**±0.1	69.7±2.0
Seismic	20.7±1.9	**30.6**±1.4	26.2±0.7	26.2±1.7
Musk	**100**±0.0	**100**±0.0	**100**±0.0	**100**±0.0
Speech	5.2±1.2	5.1±1.0	**9.3**±0.8	**10.2**±2.9
Thyroid	**76.8**±1.2	57.3±1.3	**77.0**±0.6	31.6±5.4
Abalone	**68.7**±2.3	43.4±4.8	59.7±0.1	43.2±7.0
Optdigits	66.3±10.1	**90.0**±1.2	62.0±2.7	89.0±1.0
Satimage	92.4±0.7	93.8±1.2	**94.8**±0.8	93.7±1.7
Satellite	73.2±1.6	76.3±0.4	74.6±0.7	**77.8**±0.4
Pendigits	82.3±4.5	91.0±1.4	**92.5**±1.3	**93.1**±1.2
Annthyr.	45.4±1.8	37.8±0.6	**57.7**±0.6	19.6±1.2
Mnist	**85.9**±0.0	69.4±0.9	71.8±0.3	69.7±2.0
Mammo.	29.4±1.4	38.8±1.5	**43.6**±0.5	38.7±1.7
Shuttle	**98.4**±0.1	97.3±0.2	**98.2**±0.3	95.2±0.5
Mulcross	**100**±0.0	**100**±0.0	**100**±0.0	**100**±0.0
Forest	44.0±4.1	**92.1**±0.3	58.0±10	7.6±1.2
Campaign	46.8±1.4	41.6±0.4	**49.8**±0.3	42.0±0.3
Fraud	57.9±2.8	**60.5**±1.5	58.1±3.2	41.8±1.1
Backdoor	86.6±0.1	**88.5**±0.1	84.1±0.1	10.2±0.6
mean	67.2	67.7	**68.8**	57.5
mean std	2.9	2.2	**2.0**	3.1
mean rk	3.8	5.5	**3.1**	6.1

Table 12: Mean rank (F1-score) for the experiments conducted, without Mask-KNN and with Mask-KNN

Method	mean rank	mean rank (w/ Mask-KNN)	diff.
DROCC (abalone)	10.8	11.6	+0.8
GOAD (thyroid)	7.4	8.4	+1.0
NeuTraL-AD (arrhythmia)	9.0	9.8	+0.8
Internal Cont.	3.5	3.8	+0.3
COPOD	9.7	10.3	+0.6
IForest	7.0	7.5	+0.5
KNN	4.9	5.5	+0.6
PIDForest	10.7	11.4	+0.7
RRCF	11.7	12.7	+1.0
Mask-KNN	N/A	6.1	N/A
NPT-AD	**3.0**	**3.1**	**+0.1**

C.4 Computational time

Table 13: Runtime in seconds of NPT-AD for the training and inference phase. The training runtime corresponds to the average training time of the model over the 20 runs with the parameters displayed in table 5. The inference runtime corresponds to the average runtime over the 20 runs to compute NPT-AD as shown in equation 6.

Dataset	train	inference
Wine	63	68
Lympho	10	283
Glass	76	6
Vertebral	128	2
Wbc	10	479
Ecoli	11	23
Ionosph.	12	76
Arrhyth.	100	223
Breastw	7	6
Pima	37	18
Vowels	62	63
Letter	105	15
Cardio	10	97
Seismic	9	189
Musk	56	168
Speech	62	64
Thyroid	253	2
Abalone	55	50
Optdigits	127	152
Satimage2	13	17
Satellite	13	23
Pendigits	78	47
Annthyr.	22	5
Mnist	478	153
Mammo.	16	24
Shuttle	16	115
Mullcross	43	44
Forest	73	409
Campaign	52	251
Fraud	141	362
Backdoor	18396	1992