Beyond Individual Input for Deep Anomaly Detection on Tabular Data

Hugo Thimonier [1] **Fabrice Popineau** [1] **Arpad Rimmel** [1] **Bich-Liên Doan** [1]

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 an 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 further proves 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 (Yanmin et al., 2011). Those applications include fraud detection (Hilal et al., 2022), intrusion detection in cybersecurity (Malaiya et al., 2018) or astronomy (Reyes & Estévez, 2020).

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 (Schölkopf et al., 1999; Tax & Duin, 2004; Liu et al., 2008; Ruff et al., 2018; Kim et al., 2020; Liznerski et al., 2021). However, recent work (Bergman & Hoshen, 2020; Qiu et al., 2021; Shenkar & Wolf, 2022) has revealed that the best-performing methods for tabular data involve models tailored to consider the particular structure of this data type. AD methods for structured data typically identify anomalies by using either *feature-feature* or *sample-sample* dependencies. For instance, in (Shenkar & Wolf, 2022), 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 (Thimonier et al., 2022) identifies anomalies in tabular datasets by focusing on sample-sample dependencies by measuring the influence of training *normal* samples on the validation samples. Both approaches have demonstrated competitive results for anomaly detection in tabular datasets.

Recent work on supervised deep learning methods for tabular data (Shavitt & Segal, 2018; Arik & Pfister, 2021; Gorishniy et al., 2023; Somepalli et al., 2021; Kossen et al., 2021) has also highlighted the importance of considering the particular structure of tabular data. In particular, in

[1]Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire Interdisciplinaire des Sciences du Numérique, 91190, Gif-sur-Yvette, France.. Correspondence to: Hugo Thimonier <name.surname@lisn.fr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

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

(Kossen et al., 2021; Somepalli et al., 2021; Gorishniy et al., 2023), 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 hypothesize that feature-feature relations and sample-sample dependencies are class-dependent, and **both dependencies should be used conjointly 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) (Kossen et al., 2021), first proposed for supervised tasks on tabular datasets. We show that NPTs are particularly relevant for flagging anomalies, in line with recent work (Han et al., 2022) demonstrating the effectiveness of new deep learning architectures 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 reconstruction-based approaches similar to our proposed method but utilizing K-nearest neighbors (KNN) imputation and a vanilla transformer (Vaswani et al., 2017), provides evidence that considering both types of dependencies can be crucial to detect anomalies on specific datasets accurately.

The present work offers the following contributions:

- We propose the first AD method for tabular data relying on masked feature reconstruction.
- We put forward the first deep AD 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.
- We provide strong 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 (Parzen, 1962), GMM (Roberts & Tarassenko, 1994), or Copula as in COPOD (Li et al., 2020). Other approaches also focused on local density estimation to detect outliers, such as Local Outlier Factor (LOF) (Breunig et al., 2000). In inference, one flags the samples that lie in low-probability regions under the estimated distribution as anomalies.

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 (Hawkins, 1974) or neural networks such as diverse types of autoencoders (Principi et al., 2017; Chen & Konukoglu, 2018; Kim et al., 2020), or GANs (Schlegl et al., 2017).

One-Class Classification The term *one-class classification* was coined in (Moya & Hush, 1996) and describes identifying anomalies without directly estimating the *normal* density. One-class classification (OCC) involves discriminative models that directly estimate a decision boundary. For instance, in kernel-based approaches (Schölkopf et al., 1999; Tax & Duin, 2004), 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 (Ruff et al., 2018). In (Goyal et al., 2020), 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 models such as isolation forest (IForest) (Liu et al., 2008), extended isolation forest (Hariri et al., 2021), RRCF (Guha et al., 2016) and PIDForest (Gopalan et al., 2019).

Self-Supervised Approaches Recent methods have also considered self-supervision as a means to identify anomalies. In GOAD (Bergman & Hoshen, 2020), 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 NeuTraL-AD (Qiu et al., 2021), 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, (Shenkar & Wolf, 2022) also propose a contrastive framework in which they identify samples as anomalies based on their inter-feature relations. Other self-supervised approaches, such as (Sohn et al., 2021; Reiss & Hoshen, 2023), have focused on representation learning to foster the performance of one-class classification models.

Attention Mechanisms First introduced in (Vaswani et al., 2017), 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 (Parmar et al., 2018) or image classification with the Vision Transformer (ViT) (Dosovitskiy et al., 2021), natural language processing *e.g.* Masked Language Models (MLM) such as BERT (Devlin et al., 2019), and classification tasks on structured datasets (Somepalli et al., 2021; Kossen et al., 2021).

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 (Grinsztajn et al., 2022; Shwartz-Ziv & Armon, 2021). 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 (Shavitt & Segal, 2018; Jeffares et al., 2023), 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, (Kadra et al., 2021) shows that correctly selecting a combination of regularization techniques can suffice for a Multi-Layer Perceptron (MLP) to compete with GBDT. Finally, (Somepalli et al., 2021; Kossen et al., 2021) 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 (Kossen et al., 2021), 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 learning a mapping $\phi_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ 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 (Shenkar & Wolf, 2022) employs a masking strategy that maximizes the mutual information between each sample and its masked-out part by minimizing a contrastive loss. Recently, (Kong et al., 2020) demonstrated how stochastic masking (Devlin et al., 2019) also maximizes mutual information, thereby establishing a link between the method of (Shenkar & Wolf, 2022) 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 $\mathbf{m} \in \mathbb{R}^d$ be a binary vector taking value 1 when the corresponding entry in $\mathbf{x}$ is masked, 0 otherwise. Let $\mathbf{x}^m, \mathbf{x}^o \in \mathbb{R}^d$ represent respectively the masked and unmasked entries of sample $\mathbf{x}$ defined as

$$\begin{aligned} \mathbf{x}^m &= \mathbf{m} \odot \mathbf{x} \\ \mathbf{x}^o &= (\mathbf{1}_d - \mathbf{m}) \odot \mathbf{x}, \end{aligned} \tag{2}$$

where $\mathbf{1}_d$ is the d-dimensional unit vector.

In this framework, the objective in eq. (1) is modified to

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

where $\phi_\theta(\mathbf{x}^o) \in \mathbb{R}^d$ 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 recon-

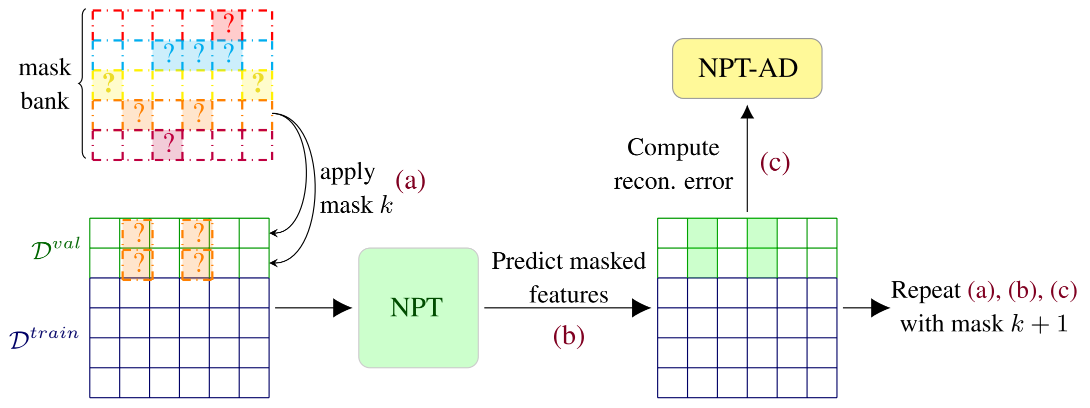


Figure 1. NPT-AD inference pipeline. In step (a), mask j is applied to each validation sample. We construct a matrix $\mathbf{X}$ composed of the masked validation samples and the whole *unmasked* training set. In step (b), we feed $\mathbf{X}$ to the Non-Parametric Transformer (NPT), which tries to 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.

struction 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. (3) can then be reformulated as

$$\min_{\theta\in\Theta} \sum_{\mathbf{x}\in\mathcal{D}_{train}} d\left(\mathbf{x}^m, \phi_\theta\left(\mathbf{x}^o \mid \mathbf{X}^O\right)\right). \qquad (4)$$

3.2. Non-parametric transformer (NPT)

We resort to Non-Parametric Transformer (NPT) (Kossen et al., 2021) 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 (Bahdanau et al., 2014; Devlin et al., 2019; Vaswani et al., 2017). We discuss MHSA more thoroughly in appendix A and only detail in this section the two mechanisms put forward in (Kossen et al., 2021).

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 relations 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 $\mathbb{R}^{n\times h}$ where $h = d \cdot e$. Then, NPT applies MHSA, as seen in equation (13) in appendix A, between the data samples flattened representations $\{\mathbf{H}_i^{(\ell)} \in \mathbb{R}^{1\times h} | i \in 1, \dots, n\}$.

$$\text{ABD}(\mathbf{H}^{(\ell)}) = \text{MHSA}(\mathbf{H}^{(\ell)}) = \mathbf{H}^{(\ell+1)} \in \mathbb{R}^{n\times h} \qquad (5)$$

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 $\mathbf{H}_i^{(\ell)} \in \mathbb{R}^{d\times e}, i \in \{1, \dots, n\}$.

$$\text{ABA}(\mathbf{H}^{(\ell)}) = \underset{\text{axis=n}}{\text{stack}}\left(\text{MHSA}(\mathbf{H}_1^{(\ell)}), \dots, \text{MHSA}(\mathbf{H}_n^{(\ell)})\right) \qquad (6)$$

3.3. Anomaly score

We directly derive the anomaly score from the loss optimized during training. The loss corresponds to the squared difference between the reconstructed feature and its actual value for numerical features. 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 , 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 a mask bank composed of 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 $\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\}$ 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

$$\text{NPT-AD}(\mathbf{z}; \mathcal{D}^{train}) = \frac{1}{m} \sum_{k=1}^{m} \mathcal{L}(\mathbf{z}^{(k)}; \mathcal{D}^{train}), \quad (7)$$

where $\mathcal{L}(\mathbf{z}^{(k)}; \mathcal{D}^{train})$ 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 (Shenkar & Wolf, 2022). 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 the literature and also omit the KDD dataset since it presents a certain number of limitations (Silva et al., 2020). Instead, we include three real-world datasets from (Han et al., 2022) that display relatively similar characteristics to KDD in terms of dimensions: fraud, campaign, and backdoor. See appendix C for more detail on the datasets' characteristics.

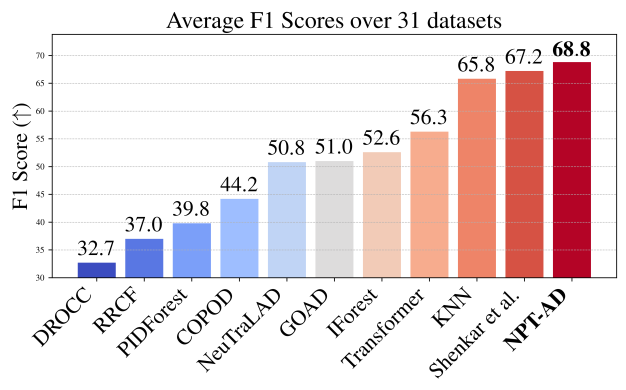
Experimental settings Per the literature (Zong et al., 2018; Bergman & Hoshen, 2020), we construct the training set with a random subsample of the *normal* samples representing 50% of the *normal* samples, we concatenate the 50% remaining with the entire set of anomalies to constitute the validation set. Following previous work, (Bergman & Hoshen, 2020), the decision threshold for the NPT-AD score is chosen such that the number of predicted anomalies is equal to the number of existing anomalies. We report the results in tables 4, 5 6 and 7 in appendix D.1. Most metrics are obtained from (Shenkar & Wolf, 2022), apart from NeuTraL-AD (Qiu et al., 2021), which we trained using their official code made available online, the transformer model, and the experiments on the fraud, campaign, and backdoor datasets. Per the literature, we evaluate the different methods using the F1-score (↑) and AUROC (↑) metrics. We compare our method to recent deep methods, namely GOAD (Bergman & Hoshen, 2020), DROCC (Goyal et al., 2020), NeuTraL-AD (Qiu et al., 2021), the contrastive approach proposed in (Shenkar & Wolf, 2022) and a transformer model (Vaswani et al., 2017) trained in a similar way to NPT-AD. We also compare to classical non-deep methods such as Isolation Forest (Liu et al., 2008), KNN (Ramaswamy et al., 2000), RRCF (Guha et al., 2016), COPOD (Li et al., 2020) and PIDForest (Gopalan et al., 2019). We refer the reader to (Shenkar & Wolf, 2022) for implementation details of non-deep models.

Notice that for DROCC (Goyal et al., 2020), GOAD (Bergman & Hoshen, 2020), and NeuTraL-AD (Qiu et al., 2021), we report in tables 4 and 5 the architecture that obtained the highest mean F1-score. The metrics obtained for the other architectures are detailed in tables 8, 9, and 10 in appendix D.1. We included each architecture of each approach to compute the mean rank as shown in 2. 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 the model can also be trained on a single GPU for small and medium datasets.

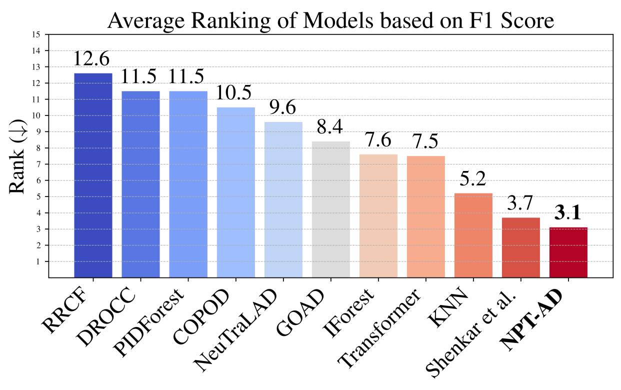
For each dataset, we considered the same NPT architecture composed of 4 layers alternating between Attention Between Datapoints and Attention Between Attributes and 4 attention heads. Per (Kossen et al., 2021), we consider a Row-wise feed-forward (rFF) network with one hidden layer, 4x expansion factor, GeLU activation, and also include dropout with $p = 0.1$ for both attention weights and hidden layers. We used LAMB (You et al., 2020) with $\beta = (0.9, 0.999)$ as the optimizer and also included a Lookahead (Zhang et al., 2019) wrapper with slow update rate $\alpha = 0.5$ and $k = 6$ steps between updates. Similarly, following (Kossen et al., 2021), 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

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

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



(a) F1-score (↑)



(b) Rank (↓)

Figure 2. For each of the 31 datasets on which models were evaluated, we compute the average F1-score over 20 runs for 20 different seeds. We report on figure 2(a) the average F1-score over all datasets for each tested model. We report on figure 2(b) the average rank over the 31 datasets. For both figures, the model displayed on the far left is the worst-performing model for the chosen metric, and on the far right is the best-performing model. We also highlight the metric of the best-performing model in bold. See tables 4 and 5 in appendix D.1 for the details of the obtained metrics.

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 appendix C 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].

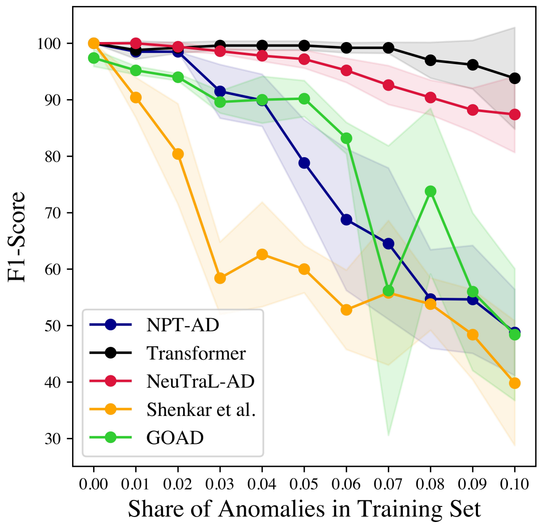
Results As seen in figure 2 and tables 4 and 5, our model surpasses existing methods on most datasets by a significant margin regarding the F1-score. Moreover, our approach displays the highest mean F1-score and lowest mean rank over all datasets out of the 18 tested approaches. The method of Shenkar & Wolf (2022) ranks as the second highest in terms of average F1-score and the second highest mean rank over all datasets, while the simple KNN-based method (Ramaswamy et al., 2000) ranks as the third best approach in terms of average F1-score and rank. Also, our approach displays a smaller variance than competing methods except for COPOD (Li et al., 2020), 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 tables 6 and 7 in appendix D.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.

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

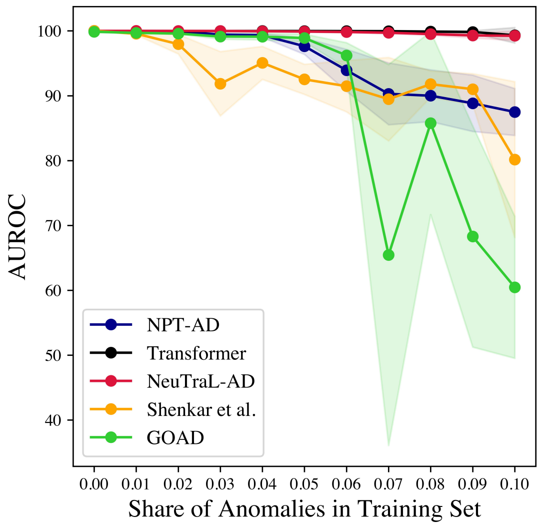
5. Discussion

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 3 in which we show how the performance of NPT-AD varies when the contamination share increases in comparison with a vanilla transformer (Vaswani et al., 2017) for mask reconstruction, NeuTraL-AD (Qiu et al., 2021), GOAD (Bergman & Hoshen, 2020) and the internal contrastive approach of (Shenkar & Wolf, 2022).



(a) F1-score (↑)



(b) AUROC (↑)

Figure 3. 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 and Transformer were trained for 100 epochs with batch size equal to the dataset size, with learning rate 0.01, optimizer LAMB (You et al., 2020) with $\beta = (0.9, 0.999)$, per-feature embedding dimension 16, r set to 1, and masking probability $p_{mask} = 0.15$. NeuTraL-AD and GOAD were trained with hyperparameters as for the thyroid dataset in the original papers and (Shenkar & Wolf, 2022) with its default parameters in their implementation.

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. 5%), 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 (Shenkar & Wolf, 2022) for both F1-score and AUROC. We also notice that the method of Shenkar & Wolf (2022) is particularly sensible to dataset contamination regarding the F1-score compared with NeuTraL-AD, GOAD, the transformer approach, and NPT-AD even for low contamination shares.

Finally, we observe that the transformer approach suffers significantly less from training set contamination than NPT-AD, especially for high contamination shares. What could account for such a difference is the fact that the contamination effect is double for NPT-AD. First, during training, the model acquires the ability to reconstruct not only normal samples but also anomalies, thereby compromising the reconstruction error as a good proxy for anomalousness. Second, in inference, the model relies on the unmasked training set to reconstruct samples: normal samples may attend to anomaly samples, which may hamper the ability of the model to reconstruct them properly. Conversely, anomalies can attend to other anomalies, which helps reduce the reconstruction error for this class.

5.2. Sample-sample dependencies ablation study

Table 1. Ablation study. Comparison in AD performance between the vanilla transformer, Mask-KNN, and NPT-AD.

	Transformer	Mask-KNN	NPT-AD
F1	56.2	57.5	**68.8**
AUROC	83.4	84.5	**89.8**

To further explore the combined impact of sample-sample and feature-feature dependencies, we introduce a reconstruction-based technique similar to NPT-AD. We put forward Mask-KNN that relies on KNN imputation to reconstruct masked features (see alg. 1 in appendix D.2). We also compare it to the vanilla transformer model trained in a framework similar to NPT-AD. Mask-KNN (resp. the transformer) can be seen as approximately equivalent to NPT-AD without considering the feature-feature dependencies (resp. the sample-sample dependencies) as illustrated in figure 4 in appendix D.2. Our experiment is detailed in appendix

D.2 and summarized in tables 1, 11 and 13. We observe that, indeed, NPT-AD outperforms the vanilla transformer and Mask-KNN based on both AUROC and F1-score as shown in table 1, emphasizing that combining both types of dependencies boosts anomaly detection performance.

Furthermore, we anticipate observing distinct performance dynamics in NPT-AD compared to both the vanilla transformer and Mask-KNN. In scenarios where the identification of anomalies relies heavily on feature-feature dependencies, such as datasets where the vanilla transformer significantly outperforms Mask-KNN, we anticipate NPT-AD to surpass Mask-KNN. This superiority is attributed to NPT-AD's capacity to effectively harness feature-feature dependencies. Conversely, in datasets where sample-sample dependencies play a pivotal in detecting anomalies, *i.e.* datasets on which Mask-KNN outperforms the vanilla transformer, we expect NPT-AD to surpass the vanilla transformer. This relationship between the performance metrics across the three approaches would validate that NPT-AD effectively integrates both feature-feature and sample-sample dependencies.

As displayed in table 11, our experiments show that we do observe this performance behavior. For instance, on datasets where Mask-KNN significantly outperforms the transformer by a sizable gap, *e.g.* glass, vertebral, or vowels, NPT-AD also outperforms the transformer significantly. Similarly, on datasets where the transformer outperforms Mask-KNN, *e.g.* lymphography, WBC or Forest Cover, NPT-AD also performs significantly better than Mask-KNN. This performance relation can help identify which datasets require which type of dependency to flag anomalies using reconstruction-based methods correctly. We propose such classification in table 12 in appendix D.2.

6. Limitations and Conclusion

Limitations As with most non-parametric models, NPT-AD displays a 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 14 in appendix D.3, 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 reconstructions 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 successfully utilize both feature-feature and sample-sample dependencies to identify anomalies. Our experiments on an extensive benchmark demonstrate 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. Finally, this work emphasizes the importance of leveraging both dependencies to effectively detect anomalies on tabular datasets.

Future Work Our work invites further research on the optimal way to combine sample-sample and feature-feature dependencies for anomaly detection on tabular data. In the current work, we relied on NPTs, which combine both dependencies *internally*. However, this restrains pretext tasks used for self-supervised AD to tasks that NPTs can perform. Other forms of *external* retrieval methods may be interesting to investigate to be able to *augment* existing AD methods.

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.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Arik, S. Ö. and Pfister, T. Tabnet: Attentive interpretable tabular learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 6679–6687. AAAI Press, 2021. doi: 10.1609/aaai.v35i8.16826. URL https://doi.org/10.1609/aaai.v35i8.16826.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate, 2014. URL https://arxiv.org/abs/1409.0473.

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

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. 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.

Chen, X. and Konukoglu, E. 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.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *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)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. 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.

Gopalan, P., Sharan, V., and Wieder, U. Pidforest: Anomaly detection and certification via partial identification. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:202766416.

Gorishniy, Y., Rubachev, I., Kartashev, N., Shlenskii, D., Kotelnikov, A., and Babenko, A. Tabr: Tabular deep learning meets nearest neighbors in 2023, 2023.

Goyal, S., Raghunathan, A., Jain, M., Simhadri, H. V., and Jain, P. Drocc: Deep robust one-class classification. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3711–3721. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/goyal20c.html.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. 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.

Guha, S., Mishra, N., Roy, G., and Schrijvers, O. Robust random cut forest based anomaly detection on streams. In *International Conference on Machine Learning*, 2016.

Han, S., Hu, X., Huang, H., Jiang, M., and Zhao, Y. AD-Bench: 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.

Hariri, S., Kind, M. C., and Brunner, R. J. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1479–1489, 2021. doi: 10.1109/TKDE.2019.2947676.

Hawkins, D. M. 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.

Hilal, W., Gadsden, S. A., and Yawney, J. 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.

Jeffares, A., Liu, T., Crabbé, J., Imrie, F., and van der Schaar, M. 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.

Kadra, A., Lindauer, M., Hutter, F., and Grabocka, J. Well-tuned simple nets excel on tabular datasets. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Kim, K. H., Shim, S., Lim, Y., Jeon, J., Choi, J., Kim, B., and Yoon, A. S. Rapp: Novelty detection with reconstruction along projection pathway. In *International Conference on Learning Representations*, 2020.

Kong, L., de Masson d'Autume, C., Yu, L., Ling, W., Dai, Z., and Yogatama, D. A mutual information maximization perspective of language representation learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Syx79eBKwr.

Kossen, J., Band, N., Lyle, C., Gomez, A., Rainforth, T., and Gal, Y. Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=wRXzOa2z5T.

Li, Z., Zhao, Y., Botta, N., Ionescu, C., and Hu, X. 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.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008. doi: 10.1109/ICDM.2008.17.

Liznerski, P., Ruff, L., Vandermeulen, R. A., Franks, B. J., Kloft, M., and Muller, K. R. Explainable deep one-class classification. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=A5VV3UyIQz.

Malaiya, R. K., Kwon, D., Kim, J., Suh, S. C., Kim, H., and Kim, I. An empirical evaluation of deep learning for network anomaly detection. In *2018 International Conference on Computing, Networking and Communications (ICNC)*, pp. 893–898, 2018. doi: 10.1109/ICCNC.2018.8390278.

Moya, M. M. and Hush, D. R. 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.

Parmar, N. J., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. Image transformer. In *International Conference on Machine Learning (ICML)*, 2018. URL http://proceedings.mlr.press/v80/parmar18a.html.

Parzen, E. 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.

Principi, E., Vesperini, F., Squartini, S., and Piazza, F. Acoustic novelty detection with adversarial autoencoders. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 3324–3330, 2017. doi: 10.1109/IJCNN.2017.7966273.

Qiu, C., Pfrommer, T., Kloft, M., Mandt, S., and Rudolph, M. Neural transformation learning for deep anomaly detection beyond images. In Meila, M. and Zhang, T. (eds.), *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*, pp. 8703–8714. PMLR, 2021. URL http://proceedings.mlr.press/v139/qiu21a.html.

Ramaswamy, S., Rastogi, R., and Shim, K. Efficient algorithms for mining outliers from large data sets. In Chen, W., Naughton, J. F., and Bernstein, P. A. (eds.), *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, pp. 427–438. ACM, 2000. doi: 10.1145/342009.335437. URL https://doi.org/10.1145/342009.335437.

Reiss, T. and Hoshen, Y. Mean-shifted contrastive loss for anomaly detection. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i2.25309. URL https://doi.org/10.1609/aaai.v37i2.25309.

Reyes, E. and Estévez, P. A. Transformation based deep anomaly detection in astronomical images. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2020. doi: 10.1109/IJCNN48605.2020.9206997.

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

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/ruff18a.html.

Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., and Langs, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In Niethammer, M., Styner, M., Aylward, S., Zhu, H., Oguz, I., Yap, P.-T., and Shen, D. (eds.), *Information Processing in Medical Imaging*, pp. 146–157, Cham, 2017. Springer International Publishing. ISBN 978-3-319-59050-9.

Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., and Platt, J. Support vector method for novelty detection. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, pp. 582–588, Cambridge, MA, USA, 1999. MIT Press.

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

Shenkar, T. and Wolf, L. Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*, 2022.

Shwartz-Ziv, R. and Armon, A. 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.

Silva, J. V. V., Lopez, M. A., and Mattos, D. M. F. 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)*, pp. 1–8, 2020. doi: 10.1109/CIoT50422.2020.9244289.

Sohn, K., Li, C.-L., Yoon, J., Jin, M., and Pfister, T. Learning and evaluating representations for deep one-class classification. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=HCSgyPUfeDj.

Somepalli, G., Goldblum, M., Schwarzschild, A., Bruss, C. B., and Goldstein, T. 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.

Tax, D. and Duin, R. Support vector data description. *Machine Learning*, 54:45–66, 01 2004. doi: 10.1023/B:MACH.0000008084.60811.49.

Thimonier, H., Popineau, F., Rimmel, A., Doan, B.-L., and Daniel, F. TracInAD: Measuring influence for anomaly detection. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, 2022. doi: 10.1109/IJCNN55064.2022.9892058. URL https://doi.org/10.1109/IJCNN55064.2022.9892058.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Yanmin, S., Wong, A., and Kamel, M. S. Classification of imbalanced data: a review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23:687–719, 11 2011. doi: 10.1142/S0218001409007326.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., and Hsieh, C.-J. 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.

Zhang, M., Lucas, J., Ba, J., and Hinton, G. E. Lookahead optimizer: k steps forward, 1 step back. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *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.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

A. Multi-Head Self-Attention

Scaled dot-product attention as first proposed in (Vaswani et al., 2017) 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

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{h_k}}\right)\mathbf{V} \tag{8}$$

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:

$$\text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \underset{\text{axis=k}}{\text{concat}}(O_1,\dots,O_k)W^O \tag{9}$$

$$\text{where, } O_j = \text{Attention}(\mathbf{Q}W_j^Q, \mathbf{K}W_j^K, \mathbf{V}W_j^V) \tag{10}$$

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:

$$\text{MHSelfAtt}(\mathbf{H}) = \text{MultiHead}(\mathbf{Q}=\mathbf{H},\mathbf{K}=\mathbf{H},\mathbf{V}=\mathbf{H}) \tag{11}$$

As described in (Kossen et al., 2021), NPT follows transformer best practices to improve performances and involves a residual branch as well as layer normalization (LN) (Ba et al., 2016) before MHSelfAtt(.).

$$\text{Res}(\mathbf{H}) = \mathbf{H}W^{\text{res}} + \text{MHSelfAtt}\left(\text{LN}(\mathbf{H})\right) \tag{12}$$

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):

$$\text{MHSA}(\mathbf{H}) = \text{Res}(\mathbf{H}) + \text{rFF}\left(\text{LN}\left(\text{Res}\left(\mathbf{H}\right)\right)\right) \in \mathbb{R}^{n\times h} \tag{13}$$

B. Training Pipeline

Let $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ be a sample with d features, which can be either numerical or categorical. Let e designate the hidden dimension of the transformer. The training pipeline consists of the following steps:

Masking We sample from a Bernouilli distribution with probability p_{mask} whether each of the d features is masked.

$$\text{mask} = (m_1,\dots,m_d),$$

where $m_j \sim \mathcal{B}(1, p_{mask})\ \forall j \in [1,...,d]$ and $m_j = 1$ if feature j is masked.

Encoding For numerical features, we normalize to obtain 0 mean and unit variance, while we use one-hot encoding for categorical features. At this point, each feature j for $j \in [1,2,...,d]$ has an e_j-dimensional representation, $encoded(\mathbf{x}_j) \in \mathbb{R}^{e_j}$, where $e_j = 1$ for numerical features and for categorical features e_j corresponds to its cardinality. We then mask each feature according to the sampled mask vector and concatenate each feature representation with the corresponding mask indicator function. Hence, each feature j has an (e_j+1)-dimensional representation

$$((1-m_j)\cdot encoded(\mathbf{x}_j), m_j) \in \mathbb{R}^{e_j+1},$$

where $\mathbf{x}_j$ is the j-th features of sample $\mathbf{x}$.

In-Embedding We pass each of the features encoded representations of sample $\mathbf{x}$ through learned linear layers $\texttt{Linear}(e_j + 1, e)$. We also learn e-dimensional index and feature-type embeddings as proposed in (Kossen et al., 2021). Both are added to the embedded representation of sample $\mathbf{x}$. The obtained embedded representation is thus of dimension $d \times e$

$$e_{\mathbf{x}} = (e_{\mathbf{x}}^1, e_{\mathbf{x}}^2, \dots, e_{\mathbf{x}}^d) \in \mathbb{R}^{d \times e},$$

and $e_{\mathbf{x}}^j \in \mathbb{R}^e$ is the embedded representation of feature j for sample $\mathbf{x}$.

NPT Encoder We concatenate the embedded representations of n samples and construct a matrix $\mathbf{X}$,

$$\mathbf{X} = (e_{\mathbf{x}_1}, \dots, e_{\mathbf{x}_n})^\top \in \mathbb{R}^{n \times d \times e}$$

that is fed to the NPT. Let $\mathbf{H} \in \mathbb{R}^{n \times d \times e}$ denote the output of the NPT,

$$\mathbf{H} = \text{NPT}(\mathbf{X}) \in \mathbb{R}^{n \times d \times e}$$

Out-Embedding The output of the NPT, $\mathbf{H} \in \mathbb{R}^{n \times d \times e}$ is then used to compute an estimation of the original d-dimensional vector for each of the n samples. To obtain the estimated feature j for the n samples, we take the j-th $n \times e$-dimensional representation which is output by the NPT encoder and pass it through a learned linear layer $\texttt{Linear}(e, e_j)$, where e_j is 1 for numerical features and the cardinality for categorical features. In other words, for each element j in the second dimension of $\mathbf{H}$, $\mathbf{H}_{:,j:}$, we compute

$$\mathbf{Z} = \mathbf{H}_{:,j:}\mathbf{W}_{out}^j \in \mathbb{R}^{n \times e_j},$$

where $\mathbf{W}_{out}^j \in \mathbb{R}^{e \times e_j}$, where $e_j = 1$ for numerical features and for categorical features e_j corresponds to its cardinality. For categorical feature, we take the $\arg\max$ along the second dimension of $\mathbf{Z}$.

C. Datasets characteristics and experimental settings

In table 2, we display the main characteristics of the datasets involved in our experiments, while in 3 we display the hyperparameters of our experiments.

Table 2. 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%)

Table 3. 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	256	0.001	0.2	1	196	32

Hyperparameter selection was done so as to obtain the fastest training loss convergence. The pipeline consisted of the following:

- **Hidden dimension** e: We started from smaller to larger models to achieve convergence.
- **Batch size**: We maximized batch size to fit into memory. Larger batch sizes are beneficial to our approach since a larger number of samples in the same batch increases the samples to which each sample can attend and fosters better learning of sample-sample dependencies.
- **Learning rate** (lr) was selected to achieve the fastest loss convergence for each architecture.
- **Masking probability** p_{mask}: we started from 0.25 and reduced with a step 0.05 until loss convergence.
- **Maximum number of features masked simultaneously** r: it was chosen in accordance with the chosen p_{mask} and the overall number of feature d. We also considered inference time when setting r as setting r higher than 1 for a dataset with many features would make inference time explode. Take the Speech dataset with $d = 400$; going from $r = 1$ to $r = 2$ would make the number of masks go from 400 to 82200.

D. Additional experiments

D.1. Additional results

In this section, we display the metrics for each of the experiments we performed. This includes the F1-score for all tested approaches in tables 4 and 5, 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 (Qiu et al., 2021) in table 10, GOAD (Bergman & Hoshen, 2020) in table 9, and DROCC (Goyal et al., 2020) in table 8. For each of these tables, we highlight in bold the highest metric in the table.

Table 4. Anomaly detection F1-score (↑) for deep models. 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. (arrhy.)	Inter.Cont.	Transformer	NPT-AD
Wine	63.0±20.0	67.0±9.4	78.2±4.5	90.0±6.3	23.5±7.9	72.5±7.7
Lympho	65.0±5.0	68.3±13.0	20.0±18.7	86.7±6.0	88.3±7.6	**94.2**±7.9
Glass	14.5±11.1	12.7±3.9	9.0±4.4	**27.2**±10.6	14.4±6.1	**26.2**±10.9
Vertebral	9.3±6.1	16.3±9.6	3.8±1.2	**26.0**±7.7	12.3±5.2	20.3±4.8
Wbc	9.0±6.2	66.2±2.9	60.9±5.6	**67.6**±3.6	66.4±3.2	**67.3**±1.7
Ecoli	N/A	61.4±31.7	7.0±7.1	70.0±7.8	75.0±9.9	**77.7**±0.1
Ionosphere	76.9±2.8	83.4±2.6	90.6±2.4	**93.2**±1.3	88.1±2.8	**92.7**±0.6
Arrhyth.	37.1±6.8	52.0±2.3	59.5±2.6	**61.8**±1.8	59.8±2.2	60.4±1.4
Breastw	93.0±3.7	96.0±0.6	91.8±1.3	96.1±0.7	**96.7**±0.3	95.7±0.3
Pima	66.0±4.1	66.0±3.1	60.3±1.4	59.1±2.2	65.6±2.0	**68.8**±0.6
Vowels	66.2±8.8	31.1±4.2	10.0±6.2	**90.8**±1.6	28.7±8.0	88.7±1.6
Letter	55.6±3.6	20.7±1.7	5.7±0.8	62.8±2.4	41.5±6.2	**71.4**±1.9
Cardio	49.8±3.2	**78.6**±2.5	45.5±4.3	71.0±2.4	68.8±2.8	**78.1**±0.1
Seismic	19.1±0.9	24.1±1.0	11.8±4.3	20.7±1.9	19.1±5.7	**26.2**±0.7
Musk	99.4±1.5	**100**±0.0	99.0±0.0	**100**±0.0	**100**±0.0	**100**±0.0
Speech	4.3±2.0	4.8±2.3	4.7±1.4	5.2±1.2	6.8±1.9	**9.3**±0.8
Thyroid	72.7±3.1	72.5±2.8	69.4±1.4	**76.8**±1.2	55.5±4.8	**77.0**±0.6
Abalone	17.9±1.3	57.6±2.2	53.2±4.0	**68.7**±2.3	42.5±7.8	59.7±0.1
Optdig.	30.5±5.2	0.3±0.3	16.2±7.3	**66.3**±10.1	61.1±4.7	62.0±2.7
Satim.	4.8±1.6	90.7±0.7	92.3±1.9	92.4±0.7	89.0±4.1	**94.8**±0.8
Satellite	52.2±1.5	64.2±0.8	71.6±0.6	73.2±1.6	65.6±3.3	**74.6**±0.7
Pendig.	11.0±2.6	40.1±5.0	69.8±8.7	82.3±4.5	35.4±10.9	**92.5**±1.3
Annthyr.	64.2±3.3	50.3±6.3	44.1±2.3	45.4±1.8	29.9±1.5	**57.7**±0.6
Mnist	N/A	66.9±1.3	84.8±0.5	85.9±0.0	56.7±5.7	**71.8**±0.3
Mammo.	32.6±2.1	33.7±6.1	19.2±2.4	29.4±1.4	17.4±2.2	**43.6**±0.5
Shuttle	N/A	73.5±5.1	97.9±0.2	**98.4**±0.1	85.3±9.8	**98.2**±0.3
Mullcr.	N/A	99.7±0.8	96.3±10.5	**100**±0.0	**100**±0.0	**100**±0.0
Forest	N/A	0.1±0.2	51.6±8.2	44.0±4.1	21.3±3.1	58.0±10
Camp.	N/A	16.2±1.8	42.1±1.7	46.8±1.4	47.0±1.9	**49.8**±0.3
Fraud	N/A	53.1±10.2	24.3±7.8	**57.9**±2.8	54.3±5.2	**58.1**±3.2
Backd.	N/A	12.7±2.9	84.4±1.8	**86.6**±0.1	85.8±0.6	84.1±0.1
mean	32.7	51.0	50.8	67.2	56.2	**68.8**
mean std	3.4	4.4	4.0	2.9	4.3	2.0
mean rk	11.5	8.4	9.6	3.7	7.5	**3.1**

Table 5. Anomaly detection F1-score (↑) for non-deep models. 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**±7.7
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	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
Optdigit	3.0±0.3	15.8±4.3	**90.0**±1.2	22.5±16.8	1.3±0.7	62.0±2.7
Satimage	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
Pendigit	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
Mullcr.	66.0±0.1	99.1±0.5	**100**±0.0	67.4±2.1	**100**±0.0	**100**±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
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	37.0	**68.8**
mean std	**1.5**	3, 9	2.2	3.6	4.0	2.0
mean rk	10.5	7.6	5.2	11.5	12.6	**3.1**

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	N/A	51.3±16	**100**±0	**100**±0
Shuttle	N/A	N/A	N/A	88.4±5.5	N/A	99.9±0.0	**100**±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	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 rk	10.3	10.0	8.7	7.4	7.7	7.2	2.9	**2.5**

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	Transformer
Wine	82.9±13.1	**95.4**±1.9	86.0±10.3	85.2±12.9	87.5±1.7	61.4±6.7
Lympho	90.7±5.1	80.9±7.5	93.1±4.1	83.1±9.9	**99.4**±0.4	**99.5**±0.4
Glass	62.2±3.0	62.9±1.2	62.5±4.6	**65.1**±2.9	63.7±3.3	61.2±7.0
Vertebral	32.8±3.5	25.8±4.5	51.9±14.9	**54.4**±16.3	32.6±1.2	44.8±5.2
Wbc	80.4±5.0	92.6±2.2	62.4±8.2	32.5±11.6	**96.3**±0.5	95.0±1.1
Ecoli	43.2±9.3	53.8±9.9	52.5±10.6	49.9±10.9	81.0±1.2	**84.8**±1.6
Iono.	87.9±2.9	**95.7**±1.7	92.9±0.9	87.2±2.7	80.3±2.1	**95.4**±1.9
Arrhyt.	76.0±1.5	80.2±1.6	77.9±1.8	76.4±2.7	80.5±1.3	**81.7**±1.1
Breastw	86.0±2.2	96.2±1.1	95.9±1.6	91.3±5.1	**99.4**±0.2	**99.6**±0.1
Pima	55.1±1.9	60.6±1.2	57.3±2.1	57.3±3.7	65.2±0.7	**67.2**±2.4
Vowels	72.3±3.1	69.7±3.8	62.2±6.1	56.1±8.0	49.6±1.0	**78.4**±9.2
Letter	40.4±3.9	35.4±0.8	36.1±2.9	37.4±4.2	50.1±0.8	**80.5**±4.8
Cardio	74.6±2.7	74.3±3.1	47.7±4.2	28.9±6.3	92.2±0.3	**93.5**±1.3
Seismic	42.7±4.2	39.9±5.5	40.6±9.4	41.0±13.1	**70.8**±0.4	58.2±7.9
Musk	99.5±0.2	99.4±0.2	98.2±1.0	91.9±3.6	94.5±0.2	**100**±0.0
Speech	46.7±1.6	48.0±1.6	52.7±3.2	**54.3**±2.3	49.1±0.5	47.2±0.7
Thyroid	**97.5**±0.2	97.1±0.2	95.8±1.6	79.4±10.4	94.1±0.2	93.8±1.2
Abalone	**92.9**±1.0	92.7±0.8	80.8±2.4	73.9±4.6	86.3±0.3	88.3±2.0
Optdigits	72.3±7.1	82.6±5.4	84.1±4.3	78.4±7.2	68.0±0.4	**96.4**±4.7
Satimage	99.6±0.4	**99.8**±0.1	**99.8**±0.1	**99.7**±0.1	97.4±0.1	**99.7**±0.1
Satellite	80.7±0.4	**81.0**±0.4	76.8±0.6	74.0±1.6	63.5±0.2	73.8±2.5
Pendigits	88.6±5.7	**98.6**±0.8	97.5±0.9	93.0±3.1	90.4±0.2	93.8±2.6
Annthyr.	**87.7**±4.2	80.2±2.9	64.9±1.6	63.7±3.0	77.4±0.4	65.4±1.4
Mnist	**97.5**±0.3	**97.8**±0.2	93.4±0.7	89.9±1.1	77.2±0.2	87.4±3.2
Mammo.	67.6±3.3	73.8±2.5	65.4±3.3	69.6±3.3	**90.5**±0.1	77.6±1.0
Mullcross	98.5±2.1	99.5±1.5	99.6±0.5	99.2±1.5	93.2±0.0	**100**±0.0
Shuttle	99.8±0.3	**99.9**±0.1	**100**±0.0	**99.9**±0.1	99.4±0.0	97.2±2.2
Forestc.	**96.8**±1.5	**96.7**±1.1	84.4±7.0	82.6±6.2	88.4±0.0	95.1±0.8
Campaign	75.4±4.8	70.0±2.1	76.5±0.7	76.4±0.5	**78.3**±0.1	75.3±2.1
Fraud	81.6±7.4	84.8±4.7	93.1±0.6	93.3±0.4	**94.7**±0.1	**94.7**±0.4
Backdoor	91.6±5.8	92.5±7.5	92.9±0.4	92.9±0.3	78.9±0.1	**95.1**±0.2
mean	77.5	79.3	76.6	72.8	79.7	**83.4**
mean std	3.5	2.5	3.6	5.1	**0.6**	2.4
mean rank	8.2	7.1	7.8	8.7	7.5	**6.1**

Table 8. DROCC (Goyal et al., 2020): 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 (Bergman & Hoshen, 2020): 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	**100**±0	**100**±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 (Qiu et al., 2021): 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±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	**60.9**±5.6	21.1±10	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±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	**99.0**±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

D.2. Mask-KNN

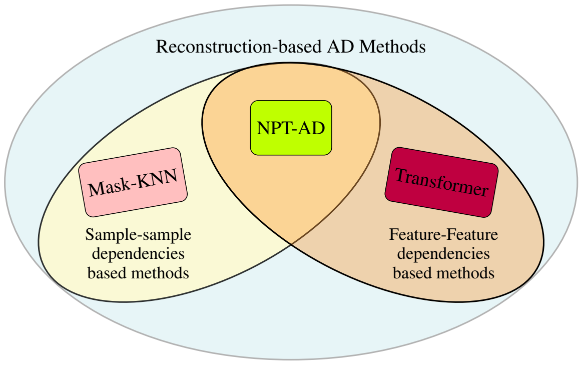

Reconstruction-based AD Methods
NPT-AD
Mask-KNN
Transformer
Sample-sample dependencies based methods
Feature-Feature dependencies based methods

Figure 4. While Mask-KNN only relies on sample-sample dependencies and the vanilla transformer only attends to feature-feature dependencies, NPT-AD combines both for anomaly detection.

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 and compare to the vanilla transformer and NPT-AD.

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 $d : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, 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 $\forall i \in \bar{\mathcal{I}}$

$$\hat{\mathbf{z}}^i = \frac{1}{k} \sum_{\mathbf{x} \in \mathcal{K}(\mathbf{z})} \mathbf{x}^i. \tag{14}$$

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

$$\hat{\mathbf{z}}^i = \frac{1}{\sum_{\mathbf{x}} \omega_{(\mathbf{z},\mathbf{x})}^{[\mathcal{I}]}} \sum_{\mathbf{x} \in \mathcal{K}(\mathbf{z})} \omega_{(\mathbf{z},\mathbf{x})}^{[\mathcal{I}]} \mathbf{x}^i. \tag{15}$$

This approach leverage only sample-sample dependencies, while the vanilla transformer only leverage feature-feature dependencies and NPT-AD combines these two types as illustrated in figure 4.

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 $d : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ a distance measure, *e.g.* the ℓ_2-norm, then the anomaly score for sample $\mathbf{z}$ is given as

$$\text{Mask-KNN}(\mathbf{z}) = \sum_{\ell=1}^{m} d(\mathbf{z}, \hat{\mathbf{z}}^{(\ell)}) \tag{16}$$

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 $\texttt{mask} \in mask\_bank$ do
    for $\texttt{idx} \in \texttt{range}(n_{val})$ do
        $\mathbf{z} \leftarrow \mathcal{D}_{val}[\texttt{idx}, :]$
        $\tilde{\mathbf{z}} \leftarrow apply\_mask(\mathbf{z}, \texttt{mask})$
        $\mathbf{X} \leftarrow (\tilde{\mathbf{z}}, B)^\top$
        $\hat{\mathbf{X}} \leftarrow$ KNNImputer$(\mathbf{X}, k)$
        $\hat{\mathbf{z}} \leftarrow \hat{\mathbf{X}}[0, :]$
        Mask-KNN$[\texttt{idx}]$ += $d(\mathbf{z}, \hat{\mathbf{z}})$
    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 (15) to reconstruct the masked sampled. We report the results of this experiment in table 11 and compare the performance of Mask-KNN, the vanilla transformer 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 13.

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 vanilla transformer 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 13.

Method	Transformer	NPT-AD	Mask-KNN
Wine	23.5±7.9	**72.5**±7.7	28.0±18.1
Lympho	88.3±7.6	**94.2**±7.9	60.0±12.2
Glass	14.4±6.1	**26.2**±10.9	**26.7**±5.4
Vertebral	12.3±5.2	20.3±4.8	**24.7**±5.9
Wbc	66.4±3.2	**67.3**±1.7	**68.1**±3.0
Ecoli	75.0±9.9	**77.7**±0.1	63.9±6.9
Ionosph.	88.1±2.8	**92.7**±0.6	89.7±0.9
Arrhyth.	59.8±2.2	60.4±1.4	**62.9**±2.4
Breastw	**96.7**±0.3	95.7±0.3	96.2±0.7
Pima	65.6±2.0	**68.8**±0.6	63.5±1.8
Vowels	28.7±8.0	**88.7**±1.6	84.3±4.9
Letter	41.5±6.2	**71.4**±1.9	56.7±3.2
Cardio	68.8±2.8	**78.1**±0.1	69.7±2.0
Seismic	19.1±5.7	**26.2**±0.7	**26.2**±1.7
Musk	**100**±0.0	**100**±0.0	**100**±0.0
Speech	6.8±1.9	**9.3**±0.8	**10.2**±2.9
Thyroid	55.5±4.8	**77.0**±0.6	31.6±5.4
Abalone	42.5±7.8	**59.7**±0.1	43.2±7.0
Optdigits	61.1±4.7	62.0±2.7	**89.0**±1.0
Satimage	89.0±4.1	**94.8**±0.8	93.7±1.7
Satellite	65.6±3.3	74.6±0.7	**77.8**±0.4
Pendigits	35.4±10.9	**92.5**±1.3	**93.1**±1.2
Annthyr.	29.9±1.5	**57.7**±0.6	19.6±1.2
Mnist	56.7±5.7	**71.8**±0.3	69.7±2.0
Mammo.	17.4±2.2	**43.6**±0.5	38.7±1.7
Shuttle	85.3±9.8	**98.2**±0.3	95.2±0.5
Mulcross	**100**±0.0	**100**±0.0	**100**±0.0
Forest	21.3±3.1	**58.0**±10	7.6±1.2
Campaign	47.0±1.9	**49.8**±0.3	42.0±0.3
Fraud	54.3±5.2	**58.1**±3.2	41.8±1.1
Backdoor	**85.8**±0.6	84.1±0.1	10.2±0.6
mean	56.2	**68.8**	57.5
mean std	4.3	**2.0**	3.1
mean rk	7.5	**3.4**	6.3

Table 12. Necessary dependencies to take into account for AD by mask-reconstruction. Given the obtained hierarchy between NPT-AD, Mask-KNN and the transformer on each dataset, we display the necessary dependencies to identify efficiently anomalies.

Dependencies	feature-feature	sample-sample
Wine	✓	✓
Lympho	✓	×
Glass	×	✓
Vertebral	×	✓
Wbc	✓	✓
Ecoli	✓	×
Ionosph.	✓	✓
Arrhyth.	✓	✓
Breastw	✓	✓
Pima	✓	✓
Vowels	×	✓
Letter	✓	✓
Cardio	✓	✓
Seismic	×	✓
Musk	✓	✓
Speech	✓	✓
Thyroid	✓	✓
Abalone	✓	✓
Optdigits	×	✓
Satimage	✓	✓
Satellite	×	✓
Pendigits	×	✓
Annthyr.	✓	✓
Mnist	×	✓
Mammo.	×	✓
Shuttle	✓	✓
Mulcross	✓	✓
Forest	✓	×
Campaign	✓	✓
Fraud	✓	×
Backdoor	✓	×

Table 13. 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)	11.5	12.4	+0.9
GOAD (thyroid)	8.4	9.1	+0.7
NeuTraL-AD (arrhythmia)	9.6	10.3	+0.7
Internal Cont.	3.7	4.0	**+0.3**
COPOD	10.5	11.0	+0.5
IForest	7.6	8.1	+0.5
KNN	5.2	5.6	+0.4
PIDForest	11.5	12.2	+0.7
RRCF	12.6	13.4	+0.8
Transformer	7.5	8.0	+0.5
Mask-KNN	N/A	6.6	N/A
NPT-AD	**3.1**	**3.4**	**+0.3**

D.3. Computational time

Table 14. 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 3. The inference runtime corresponds to the average runtime over the 20 runs to compute NPT-AD as shown in equation (7).

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