Retrieval Augmented Deep Anomaly Detection for Tabular Data

Hugo Thimonier

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

Arpad Rimmel

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

ABSTRACT

Deep learning for tabular data has garnered increasing attention in recent years, yet employing deep models for structured data remains challenging. While these models excel with unstructured data, their efficacy with structured data has been limited. Recent research has introduced retrieval-augmented models to address this gap, demonstrating promising results in supervised tasks such as classification and regression. In this work, we investigate using retrieval-augmented models for anomaly detection on tabular data. We propose a reconstruction-based approach in which a transformer model learns to reconstruct masked features of normal samples. We test the effectiveness of KNN-based and attention-based modules to select relevant samples to help in the reconstruction process of the target sample. Our experiments on a benchmark of 31 tabular datasets reveal that augmenting this reconstruction-based anomaly detection (AD) method with sample-sample dependencies via retrieval modules significantly boosts performance. The present work supports the idea that retrieval module are useful to augment any deep AD method to enhance anomaly detection on tabular data. Our code to reproduce the experiments is made available on GitHub.

CCS CONCEPTS

• Security and privacy → Intrusion/anomaly detection and malware mitigation; • Computing methodologies → Semi-supervised learning settings; *Neural networks*.

KEYWORDS

Anomaly Detection; Tabular Data; Deep Learning

ACM Reference Format:

Hugo Thimonier, Fabrice Popineau, Arpad Rimmel, and Bich-Liên Doan. 2024. Retrieval Augmented Deep Anomaly Detection for Tabular Data. In

CIKM '24, October 21-25, 2024, Boise, ID, USA.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 979-8-4007-0436-9/24/10 https://doi.org/10.1145/3627673.3679559 Fabrice Popineau fabrice.popineau@lisn.fr

Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire Interdisciplinaire des Sciences du Numérique Gif-sur-Yvette, France

Bich-Liên Doan

bich-lien.doan@lisn.fr Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire Interdisciplinaire des Sciences du Numérique Gif-sur-Yvette, France

Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3627673.3679559

1 INTRODUCTION

Semi-supervised anomaly detection (AD) consists in learning to characterize a normal distribution using a dataset only composed of samples belonging to the *normal*¹ class, in order to identify in a separate dataset the samples that do not belong to this normal distribution, namely anomalies. This class of algorithms is often used when the imbalance between classes is too severe, causing standard supervised approaches to fail [39]. Examples of such applications are cyber intrusion detection [2], fraud detection on credit card payment [15, 35], or tumor detection on images [37]. On the contrary, unsupervised anomaly detection refers to identifying anomalies in a dataset without using labeled training data. These algorithms aim to discover patterns or structures in the data and flag instances that deviate significantly from these patterns. The application of such an approach usually includes detecting mislabeled samples or removing outliers from a dataset that may hinder a model's training process.

While deep learning methods have become ubiquitous and are widely used in the industry for various tasks on unstructured data, relying on deep models for tabular data remains challenging. Indeed, Grinsztajn et al. [11] discuss how the inherent characteristics of tabular data make this type of data challenging to handle by standard deep models. Hence, recent research on deep learning for structured data [1, 9, 17, 29, 32] has been oriented towards proposing novel training frameworks, regularization techniques or architectures tailored for tabular data. Similarly, general AD methods appear to struggle with tabular data, while the best-performing AD algorithms on tabular data involve accounting for the particular structure of this data type. For instance, [3, 22, 30, 34] put forward self-supervised anomaly detection algorithms targeted for tabular data that significantly outperform general methods on most tested datasets.

In particular, recent research has emphasized the pivotal role of combining feature-feature and samples-sample dependencies in fostering deep learning model's performance on tabular data [9, 17, 32]. Following these recent findings, we investigate the benefits of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

¹The term normal relates to the concept of normality, in opposition to anomaly.

including **external retrieval modules** to leverage sample-sample dependencies to augment existing AD methods. External retrieval modules can be considered instrumental as they **can augment any existing model** that may only rely on feature-feature dependencies. In contrast, models that rely on internal retrieval mechanisms are bound to some inductive biases and cannot be used to learn all possible tasks that may be relevant for anomaly detection.

Leveraging both types of dependencies is critical to detect all types of anomalies effectively and can increase consistency across datasets, as we empirically show in section 5.1. Han et al. [13] categorize anomalies in tabular data into 4 families of anomalies which require different types of dependencies to be correctly identified. First, dependency anomalies explicitly refers to samples that do not follow the dependency structure that normal data follow require feature-feature dependencies to be efficiently identified. Second, global anomalies refer to unusual data points that deviate significantly from the norm. Relying on both dependencies should improve a model's capacity to detect these anomalies. Third, local anomalies that refer to the anomalies that are deviant from their local neighborhood can only be identified by relying on samplesample dependencies. Finally, clustered anomalies, also known as group anomalies, are composed of anomalies that exhibit similar characteristics. This type of anomaly requires feature-feature dependencies to be identified.

We test the relevance of external modules by employing transformers [36] in a mask-reconstruction framework to construct an anomaly score as it was proven to offer strong anomaly detection performance [34]. We implement several external retrieval methods to augment the vanilla transformer and evaluate the performance of each approach on an extensive benchmark of tabular datasets.

We empirically show that the tested approaches incorporating retrieval modules to account for the sample-sample relations outperform the vanilla transformer that only attends to feature-feature dependencies. Furthermore, we propose an empirical experiment to account for the pertinence of combining dependencies, showing that detecting some types of anomalies can require a particular type of dependency.

The present work offers the following contributions:

- We propose an extensive evaluation of retrieval-based methods for AD on tabular data.
- We empirically show that augmenting existing AD methods with a retrieval module to leverage sample-sample dependencies can help improve detection performance.
- We compare our approach to existing methods found in the literature and observe that our method obtains competitive performance metrics.
- We provide an explanation as to why combining dependencies leads to better identification of anomalies in tabular data.

2 RELATED WORKS

Ruff et al. [25] discuss how anomaly detection bears several denominations that more or less designate the same class of algorithms: anomaly detection, novelty detection, and outlier detection. The literature comprises 4 main classes of anomaly detection algorithms: density estimation, one-class classification, reconstruction-based, and self-supervised algorithms.

Density estimation. It is often seen as the most direct approach to detecting anomalies in a dataset. The density estimation approach consists in estimating the *normal* distribution and flagging low probability samples under this distribution as an anomaly. Existing methods include using Copula as the COPOD method proposed in [18], Local Outlier Factor (LOF) [5], Energy-based models [41] flow-based models [20].

Reconstruction-based methods. Other anomaly detection methods focus on learning to reconstruct samples belonging to the *normal* distribution. In inference, the capacity of the model to reconstruct an unseen sample is used as a measure of anomalousness. The more capable a model is to reconstruct a sample, the more likely the sample is to belong to the *normal* distribution seen in training. Such approach include methods involving autoencoders [6, 16], diffusion models [38, 42], GANs [27] or attention-based models [34].

One-Class Classification. One-class classification describes the task of identifying anomalies without *directly* estimating the *nor-mal* density. This class of algorithm involves discriminative models that directly estimate a decision boundary. In [26, 28, 33], the authors propose algorithms that estimate the support of the *normal* distribution, either in the original data space or in a latent space. During inference, one flags the samples outside the estimated support as anomalies. Other one-class classification methods include tree-based approaches such as isolation forest (IForest) [21], extended isolation forest [14], RRCF [12] and PIDForest [8]. Other methods include approaches to augment existing one-class classification methods with a classifier by generating synthetic anomalies during training, such as DROCC [10].

Self-Supervised Approaches. Given the recent successes of selfsupervision for many tasks, researchers have also investigated using self-supervised methods for anomaly detection. [3] and [22] propose transformation based anomaly detection methods for tabular data. The former relies on a classifier's capacity to identify which transformation was applied to a sample to measure anomalousness, while the latter relies on a contrastive approach. Similarly, [30] also proposes a contrastive approach to flag anomalies by learning feature-feature relation for *normal* samples. Parallel to this line of work, [24, 31] have focused on proposing self-supervised approaches for representation learning tailored for anomaly detection.

Retrieval modules. Retrieval modules have gained attention in recent years in many fields of machine learning. For instance, [4] introduces a retrieval module to foster the scalability and efficiency of diffusion models. Parallel to that, [19] introduced retrieval for cross-task generalization of large language models, and [7] introduced it to enhance prompt learning. Finally, retrieval methods have been increasingly used to increase the performance of deep models for tabular data. For instance, [17] and [32] introduced internal retrieval modules in deep architecture for supervised tasks on tabular data, while [34] relied on internal retrieval modules for anomaly detection. Finally, [9] investigated using external retrieval modules to augment an MLP for supervised tasks on tabular data.

Retrieval Augmented Deep Anomaly Detection for Tabular Data



Figure 1: Forward pass for sample z, see section 3.4 for more detail on training procedure. In the case of no retrieval module, the prediction for a sample z consists of the upper part of the figure with $\lambda = 0$.

3 METHOD

3.1 Learning Objective

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. The standard approach to anomaly detection involves learning some function $\phi_{\theta} : \mathbb{R}^d \to \mathcal{Z}$ by minimizing a loss function \mathcal{L} . The chosen loss function and the space \mathcal{Z} will vary according to the class of the considered anomaly detection algorithm. Nevertheless, the overall aim of \mathcal{L} is to characterize the distribution of the samples in the training set as precisely as possible. Depending on the chosen AD algorithm, the obtained representation $\phi_{\theta}(\mathbf{x})$ of sample \mathbf{x} can be used directly or indirectly to compute an anomaly score.

Formally, the training objective can be summarized as follows

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

where $\mathcal{L}(\mathbf{x}, \phi_{\theta}(\mathbf{x}))$ will vary according to the chosen task. For a reconstruction-based method, \mathcal{Z} can be the original data space and \mathcal{L} can be the squared ℓ_2 -norm of the difference between the original sample \mathbf{x} and its reconstructed counterpart, $\mathcal{L}(\mathbf{x}, \phi_{\theta}(\mathbf{x})) = \|\mathbf{x} - \phi_{\theta}(\mathbf{x})\|^2$.

Introducing an external retrieval module permits keeping the original objective unchanged while augmenting ϕ_{θ} with sample-sample dependencies through non-parametric mechanisms. The model involves non-parametric relations as it leverages the entire training dataset to make its prediction. Hence, the model can conjointly attend to feature-feature and sample-sample interactions to optimize its objective.

Formally, instead of minimizing the loss as described in eq. (1), the parameters θ of the function ϕ_{θ} are optimized to minimize the loss function as follows

$$\min_{\theta \in \Theta} \sum_{\mathbf{x} \in \mathcal{D}_{train}} \mathcal{L} \left(\mathbf{x}, \phi_{\theta} \left(\mathbf{x}; \mathcal{D}_{train} \right) \right).$$
(2)

Nevertheless, not all approaches to AD may benefit from such non-parametric mechanisms. Some pretext tasks involving samplesample dependencies may lead to degenerate solutions, *e.g.* approaches based on contrastive learning such as the approaches of [22] or [30]. However, reconstruction-based AD methods appear as a natural class of algorithms that may benefit from these non-parametric mechanisms.

Mask Reconstruction. In the mask reconstruction context, we empirically investigate the pertinence of external retrieval modules, as detailed in section 3.2. Our approach includes stochastic masking which consists in masking each entry in a sample vector $\mathbf{x} \in \mathbb{R}^d$ with probability p_{mask} while setting as the objective task the prediction of the masked-out features from the unmasked features. Formally, we sample $\mathbf{m} \in \mathbb{R}^d$ a binary mask vector taking value 1 when the corresponding entry in \mathbf{x} is masked, 0 otherwise. This mask \mathbf{m} is then used to construct $\mathbf{x}^m, \mathbf{x}^o \in \mathbb{R}^d$ representing respectively the masked and unmasked entries of sample $\mathbf{x}. \mathbf{x}^m, \mathbf{x}^o$ are obtained as follows,

$$\begin{aligned} \mathbf{x}^m &= \mathbf{m} \odot \mathbf{x} \\ \mathbf{x}^o &= (\mathbf{1}_d - \mathbf{m}) \odot \mathbf{x}, \end{aligned}$$
 (3)

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

A model $\phi_{\theta} : \mathbb{R}^d \times \{0, 1\}^d \to \mathbb{R}^d$ is trained to reconstruct the mask features \mathbf{x}^m from its unmasked counterpart \mathbf{x}^o and the mask vector **m**. By construction, $\phi_{\theta}(\mathbf{x}^o; \mathbf{m})$ only has non-zero values for corresponding masked features in **m**.

In the present work, we evaluate the benefit of replacing the traditional reconstruction learning objective defined as

$$\min_{\theta \in \Theta} \sum_{\mathbf{x} \in \mathcal{D}_{train}} d(\mathbf{x}^m, \phi_{\theta}(\mathbf{x}^o; \mathbf{m})), \tag{4}$$

where d(., .) is a distance measure; with its equivalent augmented with a retrieval module as follows

$$\min_{\theta \in \Theta} \sum_{\mathbf{x} \in \mathcal{D}_{train}} d\left(\mathbf{x}^{m}, \phi_{\theta}\left(\mathbf{x}^{o}; \mathbf{m}, \mathcal{D}_{train}^{o}\right)\right),$$
(5)



Figure 2: For each of the 31 datasets on which models were evaluated, we report the average F1-score over 20 runs for 20 different seeds. We refer readers to Thimonier et al. [34] for details on the obtained metrics and the hyperparameters used for each method. For both figures, the model displayed on the far left is the worst-performing model for the chosen metric, and the one on the far right is the best-performing model. We also highlight the metric of the best-performing model in bold.

where $\mathcal{D}_{train}^{o} = \{\mathbf{x}_{i}^{o} \in \mathbb{R}^{d}\}_{i=1}^{n}$. In inference, \mathcal{D}_{train}^{o} is replaced by \mathcal{D}_{train} in which none of the features of the training samples are masked.

3.2 Retrieval methods

Let z denote the sample of interest for which we wish to reconstruct its masked features z^m given its observed counterpart z^o . Let *C* denote the candidate samples from the training set from which *k helpers* are to be retrieved, and \mathcal{H} the retrieved *helpers*, $\mathcal{H} \subseteq C$.

We consider several *external* retrieval modules that rely on similarity measures to identify relevant samples to augment the encoded representation of the sample of interest z. It involves placing a retrieval module after the transformer encoder and before the output layer, as shown in figure 1. We investigate in section 5 the impact of modifying the location of the retrieval module and consider placing it before the encoder as an alternative.

For each method, the retrieval module consists in selecting the top-*k* elements that maximize a similarity measure $S(\cdot, \cdot)$ and use a value function to obtain representations of the chosen samples $\mathcal{V}(\cdot, \cdot)$ to be aggregated with sample z.

KNN-based module. First, we consider a simple method that identifies the k most relevant samples in C using a KNN approach. Formally, the similarity and value functions are defined as follows

$$\begin{split} \mathcal{S}(\mathbf{z}, \mathbf{x}) &= -\|\mathbf{h}_{\mathbf{z}} - \mathbf{h}_{\mathbf{x}}\|\\ \mathcal{V}(\mathbf{z}, \mathbf{x}) &= \mathbf{h}_{\mathbf{x}}, \end{split} \tag{6}$$

where h_x and h_z denote the representations of respectively sample x and z and $\|.\|$ is the ℓ_2 -norm.

Attention-based modules. Second, we consider attention mechanisms to select \mathcal{H} . We consider three types of attention inspired by those proposed in [9]. First, the vanilla attention (later referred to as v-attention), where the score and value function used to

select the retrieved samples are defined as

$$S(\mathbf{z}, \mathbf{x}) = W_Q(\mathbf{h}_{\mathbf{z}}) W_K(\mathbf{h}_{\mathbf{x}})$$

$$\mathcal{V}(\mathbf{z}, \mathbf{x}) = W_V(\mathbf{h}_{\mathbf{x}}).$$
(7)

where W_O , W_K and W_V are learned parameters.

Second, we also consider another type of attention module, later referred to as attention-bsim, which involves replacing the score function defined in eq. (7) as follows

$$S(\mathbf{z}, \mathbf{x}) = - \|W_K(\mathbf{h}_{\mathbf{z}}) - W_K(\mathbf{h}_{\mathbf{x}})\|^2$$

$$\mathcal{V}(\mathbf{z}, \mathbf{x}) = W_V(\mathbf{h}_{\mathbf{x}}).$$
(8)

Third, we consider attention-bsim-bval a modification of the value function in eq. (8) as

$$S(\mathbf{z}, \mathbf{x}) = - \|W_K(\mathbf{h}_z) - W_K(\mathbf{h}_x)\|^2$$

$$\mathcal{V}(\mathbf{z}, \mathbf{x}) = T(W_K(\mathbf{h}_z) - W_K(\mathbf{h}_x)),$$
(9)

where $T(\cdot) = \text{LinearWithtoutBias} \circ \text{Dropout} \circ \text{ReLU} \circ \text{Linear}(\cdot)$.

Aggregation. The retrieval modules necessitate aggregating the obtained retrieved representations $\mathcal{V}(z, \mathbf{x})$ with the representation of the sample of interest z. We aggregate the value of the selected top-*k* helpers, to be fed to the final layer

$$\tilde{\mathbf{h}}_{\mathbf{z}} = (1 - \lambda) \cdot \mathbf{h}_{\mathbf{z}} + \lambda \cdot \frac{1}{k} \sum_{\mathbf{x} \in \mathcal{H}} \mathcal{V}(\mathbf{z}, \mathbf{x}).$$
(10)

where $\lambda \in [0, 1)$ is a hyperparameter.

3.3 Anomaly score

We construct an anomaly score to assess whether a test sample belongs to the *normal* distribution or should be considered an anomaly. As a reconstruction-based method, our anomaly score is directly obtained from the optimized loss during training: the better the trained model reconstructs a sample, the more likely the sample is to be *normal*. Indeed, since the model has exclusively seen *normal* samples during training, it should be less able to reconstruct anomalies

Table 1: Comparison of transformer-based methods. We observe that the external retrieval module attention-bsim significantly improves the AD performance of the vanilla transformer by 4.3% regarding the F1-Score and 1.2% for the AUROC.

	Transformer	+KNN	+v-att.	+att-bsim	+att-bsim-bval
F1-Score (↑)	56.2	56.1	55.7	58.6	53.9
AUROC (↑)	83.4	83.1	83.1	84.4	82.1

correctly since they stem from a different distribution. On the contrary, unseen *normal* samples should be well reconstructed. We rely on the squared ℓ_2 -norm of the difference between the reconstructed sample and the original sample for numerical features, while we use the cross-entropy loss function for categorical features.

We rely on a mask bank composed of *m d*-dimensional masks to construct the anomaly score. We apply each mask to each validation sample and reconstruct the masked features to compute the reconstruction error for each mask. Thus, each validation sample is masked and reconstructed m times. The anomaly score is constructed as the average reconstruction error over the *m* masks. To construct the mask bank, we fix the maximum of features to be masked simultaneously r and and construct $m = \sum_{k=1}^{r} {d \choose k}$ masks. Choosing deterministic masks instead of random masks to create the mask bank used for inference is beneficial for two reasons. First, since the model will reconstruct all features at least once, it increases the likelihood of identifying different types of anomalies. Indeed, anomalies that deviate from the normal distribution due to a single feature would only be identified if the corresponding mask hiding this feature would be included. Second, this approach ensures that all samples are masked identically to build the anomaly score. We investigate the impact of constructing a random mask bank instead of a deterministic mask bank in section 5.5.

We use the whole unmasked training set² $C = D^{train}$ to predict the masked features of each sample for each of the *m* masked vectors and construct the anomaly score for a validation sample z as

AD-Score(
$$\mathbf{z}; \mathcal{D}^{train}$$
) = $\frac{1}{m} \sum_{k=1}^{m} \mathcal{L}(\mathbf{z}^{(k)}; \mathcal{D}^{train}),$ (11)

where $\mathcal{L}(\mathbf{z}^{(k)}; \mathcal{D}^{train})$ designates the loss for the sample \mathbf{z} with mask k.

3.4 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 Bernoulli distribution with probability p_{mask} whether each of the *d* features is masked.

$$mask = (m_1, \ldots, m_d),$$

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

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

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 **x** through learned linear layers Linear $(e_j + 1, e)$. We also learn *e*-dimensional index and feature-type embeddings as proposed in [17]. Both are added to the embedded representation of sample **x**. The obtained embedded representation is thus of dimension $d \times e$

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

and $h_{\mathbf{x}}^{j} \in \mathbb{R}^{e}$ corresponds to the embedded representation of feature *j* of sample **x**.

Transformer Encoder. The embedded representation obtained from the previous step is then passed through a transformer encoder. The output of the transformer $\mathbf{h}_{\mathbf{x}}$ is of dimension $d \times e$.

Out-Embedding. The output of the transformer, $\mathbf{h}_{\mathbf{x}} \in \mathbb{R}^{d \times e}$ is then used to compute an estimation of the original *d*-dimensional vector. To obtain the estimated feature *j*, we take the *j*-th *d*-dimensional representation which is output by the transformer encoder, $h_{\mathbf{x}_j} \in \mathbb{R}^d$, and pass it through a linear layer Linear (*e*, *e_j*), where *e_j* is 1 for numerical features and the cardinality for categorical features.

External Retrieval Modules. During training, for a batch \mathcal{B} composed of *b* samples, for each sample $\mathbf{x} \in \mathcal{B}$, the entire batch serves as the candidates *C*. In inference, a random subsample of the training set is used as *C*. Both for training and inference, when possible memory-wise, we use as \mathcal{B} and *C* the entire training set.

As input, the retrieval module receives a $\mathbb{R}^{d \times e}$ representation for each sample. Operations described in eq. (6), (7), (8) and (9) are performed on the flattened representations of samples **x**, $\mathbf{h}_{flatten}^{\mathbf{x}} \in \mathbb{R}^{d \cdot e}$. After selecting $\mathcal{H} \subseteq C$, each sample is transformed back to its original dimension to allow aggregation as described in eq. (10).

4 EXPERIMENTS

Datasets. We experiment on an extensive benchmark of tabular datasets following previous work [30, 34]. The benchmark is comprised of two datasets widely used in the anomaly detection

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

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 also include three real-world datasets from [13]: fraud, campaign, and backdoor. We display each dataset's characteristics in table 7 in appendix A.1.

Experimental Settings. Following previous work in the AD literature, [3, 44], 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. Similarly, we fix the decision threshold for the AD score such that the number of predicted anomalies equals the number of existing anomalies.

To evaluate to which extent sample-sample dependencies are relevant for anomaly detection, we compare models that attend to relations between samples to the vanilla transformer model. We compare the different methods discussed using the F1-score (\uparrow) and AUROC (\uparrow) metrics following the literature. For each dataset, we report an average over 20 runs for 20 different seeds; we display the detailed results for all five transformer-based methods in tables 8 and 9 in appendix B.1 and report in Table 1 the average F1 and AUROC.

We considered three regimes for the transformer dimensions depending on the dataset size. The transformer encoder comprises 2 or 4 layers with 4 attention heads and hidden dimension $e \in \{8, 16, 32\}$ for smaller to larger datasets. We train the transformer with a mask probability p_{mask} set to 0.25 or 0.15 and rely on the LAMB optimizer [40] with $\beta = (0.9, 0.999)$ and also included a Lookahead [43] wrapper with slow update rate $\alpha = 0.5$ and k = 6 steps between updates. We also include a dropout regularization with p = 0.1for attention weights and hidden layers. We ensure that during training, all samples from a batch are not masked simultaneously so that the retrieval module receives encoded representations of unmasked samples as it will in the inference stage. We considered the same transformer architecture and hyperparameters for the same dataset for each considered approach. For external retrieval modules, we chose for simplicity $\lambda = 0.5$ for aggregation as detailed in eq. (10). We study in section 5.3 the effect of varying the value of λ on the model's performance. For the KNN module, we set k = 5 as the cardinality of ${\mathcal H}$ since KNN-based anomaly detection methods [23] with k = 5 have shown strong anomaly detection performance [30]. In contrast, for the attention modules, we set $\mathcal{H} = C$ and use the attention weights to compute a weighted mean to be aggregated as in eq. (10). We further discuss the choice of k in section 5.2. Finally, depending on the dataset, we trained the model until the loss stopped improving for 50 or 100 consecutive epochs. Each experiment in the present work can be replicated using the code made available on github⁴.

Results. As reported in Table 1, we observe that not all retrieval modules induce a significant boost in anomaly detection performance. We observe that only the transformer augmented by the attention-bsim module performs significantly better than the vanilla transformer. Indeed, augmenting the vanilla transformer

with the retrieval module detailed in eq. (8) allows to increase the average F1-score by 4.3% and AUROC by 1.2%. This result is all the more interesting since it contradicts the results obtained for the supervised classification and regression tasks investigated in previous work [9] where the module that obtains the best performance is attention-bsim-bval. However, let us mention that the attention-bsim-bval module involved in our work is not identical to the one put forward in [9] as it does not involve any label.

For completeness, we compare the different architectures proposed in the present work to existing methods in the literature. We rely on the experiments conducted in [30, 34] for the metrics of the competing methods. We display in figure 2 the comparison to existing methods. We compare our methods to recent deep methods, namely GOAD [3], DROCC [10], NeuTraL-AD [22], the contrastive approach proposed in [30] and NPT-AD [34]. We also compare to classical non-deep methods such as Isolation Forest [21], KNN [23], RRCF [12], COPOD [18] and PIDForest [8]. We refer the reader to [34] for the detail on the F1-score per dataset for other methods than those shown in Tables 8 and 9 in appendix B.1.

5 DISCUSSION

5.1 Why combine dependencies?

To account for the fact that the retrieval-augmented transformer outperforms the vanilla transformer, we hypothesize that *different types of anomalies require different dependencies to identify them.* In this section, we provide a simple example to demonstrate this statement. Consider a simple three dimensional data space, $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$, in which the relation between the features of *normal* sample are defined as follows,

$$\begin{aligned} x_2 &= & \alpha_1 + \beta_1 x_1 + \varepsilon \\ x_3 &= & \alpha_2 + \beta_2 x_2^2 + \varepsilon, \end{aligned}$$
 (12)

where ε is some white noise and $(\alpha_1, \alpha_2, \beta_1, \beta_2) \in \mathbb{R}^2$ are scalars.

Let us consider two types of anomalies as shown in Figure 3. First anomalies of type 1, in which the relations between features are identical to the ones given in eq. (12) but in a different subspace. Now consider type 2 anomalies, for which the values of the generating feature x_1 are in the same subspace as *normal* samples, the relation between x_1 and x_2 is the same, but the parameters of the relation between x_2 and x_3 differ. Type 1 (resp. 2) anomalies are akin to *local anomalies* (resp. *dependency anomalies*) discussed in [13].

To test our hypothesis, we compare the retrieval augmented transformer to the vanilla transformer and Mask-KNN, a reconstructionbased technique introduced in [34], that relies on KNN imputation to reconstruct masked features. Mask-KNN (resp. the transformer) can be considered approximately equivalent to the retrieval augmented transformer without considering the feature-feature dependencies (resp. the sample-sample dependencies).

In the present framework, models only leveraging inter-feature relations, such as the vanilla transformer, may have limited capacities to identify anomalies if they satisfy the same relations as given in eq. (12) but in a different subspace, i.e., anomalies of type 1. Similarly, a model that only relies on inter-sample relations, e.g., Mask-KNN, would struggle to correctly identify anomalies of type 2 as they lie in a subspace close to normal samples.

³http://odds.cs.stonybrook.edu/

⁴https://github.com/hugothimonier/Retrieval-Augmented-Deep-Anomaly-Detection-for-Tabular-Data

Table 2: Comparison of the F1-score (\uparrow) of transformer+attention-bsim across values of k. Here -1 stands for $\mathcal{H} = C$. Some values are N/A either because it is not relevant to compute or when there are not enough samples in the training set for the selected value of k.

k	0	5	25	50	200	500	-1
		trans	former+att	tention-bs	im		
Abalone	42.5±7.8	$53.0 {\pm} 6.4$	54.9 ±5.4	55.0 ±5.4	52.0 ± 5.6	54.0 ± 6.5	53.0 ± 5.7
Satellite	65.6±3.3	71.5 ± 2.4	71.3±1.3	71.2 ± 1.6	70.8 ± 1.8	71.2 ± 1.7	71.9±1.5
Lympho	88.3±7.6	91.7 ± 8.3	93.3±8.2	91.7 ± 8.3	N/A	N/A	90.0 ± 8.1
Satimage	$89.0 {\pm} 4.1$	88.8 ± 3.8	88.4 ± 3.8	$89.4 {\pm} 4.2$	88.8 ± 4.3	89.1 ± 4.3	93.2 ± 1.7
Thyroid	55.5 ± 4.8	56.9±5.3	55.9 ± 5.2	56.3 ± 5.2	56.4 ± 5.2	55.9 ± 5.6	55.8 ± 6.3
Cardio	81.0 ± 4.1	81.2 ± 1.6	81.9 ± 1.4	81.9 ± 1.4	81.9 ± 1.4	81.9 ± 1.4	80.6 ± 2.4
Ionosphere	88.1±2.8	89.4 ± 5.0	90.2 ± 4.5	$89.8 {\pm} 4.3$	N/A	N/A	91.7±2.1
mean	70.3	76.1	76.6	76.5	70.0	70.4	76.6
mean std	4.9	4.7	4.3	4.3	3.7	3.9	4.0

Table 3: Share (%) of each class correctly identified (†). Average over 5 data splits. The Table should be read as follows: On average, the transformer correctly classified 78.5% of type 1 anomalies as anomalies.

	Normal	Anomalies (type 1)	Anomalies (type 2)
Mask-KNN	93.0% (±0.4)	100.0%(±0.0)	77.3%(±4.4)
Transformer	91.4% (±1.4)	78.5%(±1.0)	$100.0\%(\pm 0.0)$
+att-bsim	$94.6\% (\pm 0.5)$	$88.0\%(\pm 0.8)$	$100.0\%(\pm 0.0)$

Synthetic Dataset. We effectively test this hypothesis by constructing a synthetic three-dimensional dataset where the features of normal samples satisfy the relations described in eq. (12). We also construct two types of anomalies where type 1 anomalies follow the same inter-feature relation as normal samples but with values in a non-overlapping interval as those of the normal class. Type 2 anomalies are constructed to be close to the normal population but with inter-feature relation differing from eq. (12). This synthetic dataset is displayed in figure 3. The normal population comprises 1000 samples, and we generate 200 anomalies for each type. We use half of the normal samples to train models and use the rest merged with the anomalies as the validation set. We compare the capacity of Mask-KNN, the vanilla transformer, and the retrieval-augmented transformer to identify anomalies and display the obtained results in Table 3. We consider the same mask bank, {(1, 0, 0), (0, 1, 0), (0, 0, 1)}, for all three approaches. We use the same strategy for selecting the decision threshold as detailed in section 4. See appendix B.2 for more details on the experimental setting.

Results. We observe in Table 3 that the vanilla transformer struggles to detect type 1 anomalies in comparison with other methods, as they explicitly require inter-sample dependencies to be identified. Nevertheless, it correctly identifies 91.4% of normal samples on average and perfectly detects type 2 anomalies. Conversely, Mask-KNN obtains significantly lower performance than competing methods



Figure 3: Anomalies of type 1 (•) require inter-sample dependencies to be correctly identified with high probability. Anomalies of type 2 (•) on the other hand require interfeature dependencies to be correctly identified.

for type 2 anomalies while correctly identifying type 1 anomalies and normal samples. Notice how both methods cannot correctly identify anomalies from one of the two anomaly types despite using a perfectly separable dataset. On the contrary, the retrieval augmented transformer can better identify both types of anomalies as it can leverage inter-sample and inter-feature dependencies. This experiment provides information as to why combining dependencies for anomaly detection is relevant: it allows the detection of most anomaly types while other approaches are confined to some anomaly categories.

5.2 Analysis of the effect of the number of helpers k on the performance

Note that we randomly selected a subset of the dataset from the benchmark for the ablation studies conducted hereafter. We use this

Table 4: Comparison of the performance of the transformer + attention-bsim model for different retrieval module architecture based on the F1-score ([↑]).

Retrieval Aggregation	post-enc post-enc	post-emb post-enc	post-emb post-emb
Abalone	53.0±5.7	47.0 ± 7.6	30.5 ± 14
Satellite	71.9±1.5	60.5 ± 0.9	65.1 ± 2.3
Lympho	90.0 ±8.1	91.6±8.3	84.2 ± 11.1
Satimage	93.2 ± 1.7	50.8 ± 17.0	65.9 ± 15.3
Thyroid	55.8 ± 6.3	55.2 ± 5.9	58.1 ± 5.0
Ionosphere	91.7±2.1	88.2±1.4	91.3 ±2.0
mean	75.9	65.6	65.9
mean std	4.2	6.9	8.3

subset of datasets in each experiment for computational reasons and to avoid cherry-picking the hyperparameters.

We investigate the impact of varying k, the number of *helpers*, for both the transformer augmented by attention-bsim and the KNN module since they are the two best-performing retrieval-augmented models. We keep hyperparameters constant and only make k vary between runs. We report in Table 2 the obtained results for both architectures for values of $k \in \{0, 5, 25, 50, 200, 500, -1\}$, where -1implies that $\mathcal{H} = C$.

A noticeable trend is that both architectures obtain the best results with moderate numbers of helper, *i.e.* $k \in \{25, 50\}$, and have worse performance for smaller values of k. This observation suggests that performance displayed in tables 1,8, and 9 could be improved for optimized values of k.

5.3 Analysis of the effect of the value of λ

We analyze the performance variation for the best-performing external retrieval module, namely transformer+attention-bsim, for varying values of λ . We compute the average F1-score over 10 runs for each value of λ in {0.1, 0.2, ..., 0.7} while keeping the same hyperparameters. We report the results in Table 5.

We observe a slight variation of the average metric across the datasets when setting lambda values from 0.1 to 0.7. The maximum value is obtained for 0.5, but the difference with other values of λ is non-significant. Nevertheless, we observe significant differences between the obtained results for isolated datasets for different values of λ . This observation supports the idea that an optimal value of λ exists for each dataset, which may differ between datasets.

5.4 Location of the retrieval module

We investigate the impact of the retrieval module's location on the retrieval-augmented models' performance. To do so, we focus on the transformer+attention-bsim model since it has shown to be the best-performing retrieval-based method. We compare three different architectures:

• (post-enc, post-enc) for *post-encoder* location and *post-encoder* aggregation: the architecture detailed in figure 1,

Table 5: Comparison of transformer+attention-bsim across values of λ based on the F1-Score (\uparrow).

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Abalone	42.5	44.3	47	47.3	46.5	53	46.5	45.8
Satellite	65.6	72	72	72.5	73	71.9	73.5	73.4
Lympho	88.3	90.8	90.8	91.7	92.5	90	91.7	90
Satim.	89	94.5	94.6	94.1	94.1	93.2	93.8	94.1
Thyroid	55.5	57.2	57.4	56.6	56.6	55.8	57.1	57.3
Cardio	68.8	80.9	80.6	80.9	80.9	80.9	80.9	80.7
Ionosp.	88.1	90.3	92.1	92.2	91.2	91.7	92.3	91.7
mean	71.1	75.7	76.4	76.5	76.4	76.6	76.5	76.1

- (post-emb, post-enc), the architecture in which the retrieval module is located after the embedding layer, but the aggregation is still located after the encoder,
- (post-emb, post-emb) the architecture where both retrieval and aggregation are located after the embedding layer.

We do not report the results for the cardio dataset since it failed to converge for the (post-emb, post-enc) and (post-emb, post-emb) architectures and output NaN values in inference. We used the same hyperparameters for all three architectures. For (post-emb, post-enc) and (post-emb, post-emb) architectures we report the average over 10 runs. We display the results in Table 4.

We observe that the (post-enc, post-enc) architecture obtains the highest mean and lowest mean standard deviation over the tested datasets by a sizable margin. The transformer encoder's expressiveness allows for better representations of a data sample than the embedding layer and may account for such results. Indeed, this shows that the retrieval modules that receive the embedded representation as inputs are less able to select the relevant sample to foster mask reconstruction and anomaly detection performance. Moreover, since the encoder and retrieval modules are trained conjointly, the retrieval module can help the encoder converge to a state that favors sample representations that allow relevant clusters to be formed.

5.5 Random mask bank

We also investigate the impact of constructing a random mask bank for inference instead of selecting a deterministic mask bank as discussed in section 3.3. We construct for inference a random mask bank composed of the same number of masks as for the deterministic mask bank and use the same probability p_{mask} as used to train the model. We compare the performance of the transformer model+attention-bsim based on the F1-score for the two set-ups and display the results in Table 6. We observe that the deterministic mask bank detects anomalies better on most tested datasets. When computing the anomaly score with the deterministic mask bank, the model obtains an average F1-score of 76.6 over the 7 datasets, while with the random mask bank, the model obtains 65.4. Moreover, as expected, we also observed a significantly larger standard deviation between runs. We might expect the standard deviation to decrease as the number of masks increases, which would induce significant computational overhead.

Retrieval Augmented Deep Anomaly Detection for Tabular Data

Table 6: Comparison of the performance of the transformer + attention-bsim model based on the F1-Score (\uparrow) for two mask bank set-ups. The same model was used for inference in both set-ups. We report an average over 10 different splits of the data.

Mask bank	random	deterministic
Abalone	43.0±14.9	53.0 ±5.7
Satellite	58.4 ± 5.5	71.9±1.5
Lympho	86.7 ± 8.5	90.0 ±8.1
Satimage	47.7 ± 20.5	93.2±1.7
Thyroid	55.6 ± 6.1	55.8±6.3
Cardio	81.3 ± 1.6	80.6 ± 2.4
Ionosphere	85.2 ± 4.3	91.7±2.1
mean	65.4	76.6
mean std	8.8	4.0

6 LIMITATIONS AND CONCLUSION

Limitations. As with most non-parametric models that leverage the training set in inference, our retrieval-augmented models display a higher complexity than parametric approaches. These approaches can scale well for datasets with a reasonable number of features *d*; however, for large values of *d*, these models incur a high memory cost.

Conclusion. In this work, we have proposed an extensive investigation into external retrieval to augment reconstruction-based anomaly detection methods for tabular data. We have shown that augmenting existing AD methods using attention-based retrieval modules can help foster performance by allowing the model to attend to sample-sample dependencies. Indeed, our experiments involving an extensive benchmark of tabular datasets demonstrate the effectiveness of retrieval-based approaches since the architecture involving the attention-bsim module surpasses the vanilla transformer by a significant margin. We also provide a first explanation as to why combining both types of dependencies can be critical to obtaining consistent performance across datasets: different types of anomalies require different types of dependencies to be efficiently detected.

Future Work. Overall, our work showed that the best-performing attention-based retrieval mechanism relies on other forms of attention than vanilla attention. Parallel to that, models like NPT-AD have shown strong performance for anomaly detection on tabular data and rely on standard multi-head self-attention through the Attention Between Datapoints (ABD) mechanism, close to the v-attention module, to leverage inter-sample relations. Our findings may invite further research on modifying the ABD mechanism involved in NPTs to improve their AD performance. Finally, the use of external retrieval modules proved effective for the task of anomaly detection using mask reconstruction. The proposed external retrieval module could also be easily added to existing deep-AD methods to test whether they may prove relevant for other pretext tasks for anomaly detection 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 Centrale-Supelec and sponsored by the LUSIS company.

REFERENCES

- Sercan Ö. Arik and Tomas Pfister. 2021. 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. AAAI Press, 6679–6687. https: //doi.org/10.1609/aaai.v35i8.16826
- [2] Rachid Ben Said, Zakaria Sabir, and Iman Askerzade. 2023. CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With Hybrid Feature Selection. *IEEE Access* 11 (2023), 138732–138747. https://doi.org/10.1109/ACCESS.2023.3340142
- [3] Liron Bergman and Yedid Hoshen. 2020. Classification-Based Anomaly Detection for General Data. In International Conference on Learning Representations. https://openreview.net/forum?id=H1lK_lBtvS
- [4] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. 2022. Retrieval-Augmented Diffusion Models. In Advances in Neural Information Processing Systems, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 15309–15324. https://proceedings.neurips.cc/paper_files/paper/2022/file/ 62868cc2fc1eb5cdf321d05b4b88510c-Paper-Conference.pdf
- [5] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. SIGMOD Rec. 29, 2 (may 2000), 93–104. https://doi.org/10.1145/335191.335388
- [6] Xiaoran Chen and Ender Konukoglu. 2018. Unsupervised Detection of Lesions in Brain MRI using constrained adversarial auto-encoders. In Medical Imaging with Deep Learning. https://openreview.net/forum?id=H1nGLZ2oG
- [7] Xiang Chen, Lei Li, Ningyu Zhang, Xiaozhuan Liang, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Decoupling Knowledge from Memorization: Retrieval-augmented Prompt Learning. In Advances in Neural Information Processing Systems, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 23908–23922. https://proceedings.neurips.cc/paper_files/paper/2022/file/ 97011c648eda678424f9292dadeae72e-Paper-Conference.pdf
- [8] Parikshit Gopalan, Vatsal Sharan, and Udi Wieder. 2019. PIDForest: Anomaly Detection and Certification via Partial Identification. In Neural Information Processing Systems. https://api.semanticscholar.org/CorpusID:202766416
- [9] Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev, Daniil Shlenskii, Akim Kotelnikov, and Artem Babenko. 2023. TabR: Tabular Deep Learning Meets Nearest Neighbors in 2023. arXiv:2307.14338 [cs.LG]
- [10] Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. 2020. DROCC: Deep Robust One-Class Classification. In Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119), Hal Daumé III and Aarti Singh (Eds.). PMLR, 3711– 3721. https://proceedings.mlr.press/v119/goyal20c.html
- [11] Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. 2022. 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.* https://openreview.net/forum?id=Fp7_phQszn
- [12] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. 2016. Robust Random Cut Forest Based Anomaly Detection on Streams. In International Conference on Machine Learning.
- [13] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. 2022. ADBench: Anomaly Detection Benchmark. In *Thirty-sixth Conference on Neu*ral Information Processing Systems Datasets and Benchmarks Track. https: //openreview.net/forum?id=foA_SFQ9zo0
- [14] Sahand Hariri, Matias Carrasco Kind, and Robert J. Brunner. 2021. Extended Isolation Forest. *IEEE Transactions on Knowledge and Data Engineering* 33, 4 (2021), 1479–1489. https://doi.org/10.1109/TKDE.2019.2947676
- [15] Waleed Hilal, S. Andrew Gadsden, and John Yawney. 2022. Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances. Expert Systems with Applications 193 (2022), 116429. https://doi.org/10.1016/j.eswa.2021.116429
- [16] Ki Hyun Kim, Sangwoo Shim, Yongsub Lim, Jongseob Jeon, Jeongwoo Choi, Byungchan Kim, and Andre S. Yoon. 2020. RaPP: Novelty Detection with Reconstruction along Projection Pathway. In *International Conference on Learning Representations*.

CIKM '24, October 21-25, 2024, Boise, ID, USA.

Hugo Thimonier, Fabrice Popineau, Arpad Rimmel, and Bich-Liên Doan

- [17] Jannik Kossen, Neil Band, Clare Lyle, Aidan Gomez, Tom Rainforth, and Yarin Gal. 2021. Self-Attention Between Datapoints: Going Beyond Individual Input-Output Pairs in Deep Learning. In Advances in Neural Information Processing Systems, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.). https://openreview.net/forum?id=wRXz0a2z5T
- [18] Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. 2020. COPOD: Copula-Based Outlier Detection. In 2020 IEEE International Conference on Data Mining (ICDM). IEEE. https://doi.org/10.1109/icdm50108.2020.00135
- [19] Bill Yuchen Lin, Kangmin Tan, Chris Miller, Beiwen Tian, and Xiang Ren. 2022. Unsupervised Cross-Task Generalization via Retrieval Augmentation. In Advances in Neural Information Processing Systems, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 22003–22017. https://proceedings.neurips.cc/paper_files/paper/2022/file/ 8a0d3ae989a382ce6e50312bc35bf7e1-Paper-Conference.pdf
- [20] Boyang Liu, Pang-Ning Tan, and Jiayu Zhou. 2022. Unsupervised Anomaly Detection by Robust Density Estimation. Proceedings of the AAAI Conference on Artificial Intelligence 36, 4 (Jun. 2022), 4101–4108. https://doi.org/10.1609/aaai. v36i4.20328
- [21] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In 2008 Eighth IEEE International Conference on Data Mining. 413–422. https: //doi.org/10.1109/ICDM.2008.17
- [22] Chen Qiu, Timo Pfrommer, Marius Kloft, Stephan Mandt, and Maja Rudolph. 2021. Neural Transformation Learning for Deep Anomaly Detection Beyond Images. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139), Marina Meila and Tong Zhang (Eds.). PMLR, 8703–8714. http://proceedings.mlr. press/v139/qiu21a.html
- [23] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient Algorithms for Mining Outliers from Large Data Sets. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA, Weidong Chen, Jeffrey F. Naughton, and Philip A. Bernstein (Eds.). ACM, 427–438. https://doi.org/10.1145/342009.335437
- [24] Tal Reiss and Yedid Hoshen. 2023. 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, Article 240, 8 pages. https://doi.org/10.1609/aaai.v37i2.25309
- [25] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller. 2021. A Unifying Review of Deep and Shallow Anomaly Detection. Proc. IEEE 109, 5 (May 2021), 756–795. https://doi.org/10/gjmk3g arXiv:2009.11732
- [26] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep One-Class Classification. In Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80). PMLR, 4393–4402. http://proceedings.mlr.press/v80/ruff18a.html
- [27] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. 2017. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. In *Information Processing in Medical Imaging*, Marc Niethammer, Martin Styner, Stephen Aylward, Hongtu Zhu, Ipek Oguz, Pew-Thian Yap, and Dinggang Shen (Eds.). Springer International Publishing, Cham, 146–157.
- [28] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. 1999. Support Vector Method for Novelty Detection. In Proceedings of the 12th International Conference on Neural Information Processing Systems (Denver, CO) (NIPS'99). MIT Press, Cambridge, MA, USA, 582–588.
- [29] Ira Shavitt and Eran Segal. 2018. Regularization Learning Networks: Deep Learning for Tabular Datasets. In Advances in Neural Information Processing Systems, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/ 2018/file/500e75a036dc2d7d2fec5da1b71d36cc-Paper.pdf

- [30] Tom Shenkar and Lior Wolf. 2022. Anomaly Detection for Tabular Data with Internal Contrastive Learning. In International Conference on Learning Representations.
- [31] Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. 2021. Learning and Evaluating Representations for Deep One-Class Classification. In International Conference on Learning Representations. https://openreview.net/ forum?id=HCSgyPUfeDj
- [32] Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C. Bayan Bruss, and Tom Goldstein. 2021. SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training. *CoRR* abs/2106.01342 (2021). arXiv:2106.01342 https://arxiv.org/abs/2106.01342
- [33] David Tax and Robert Duin. 2004. Support Vector Data Description. Machine Learning 54 (01 2004), 45–66. https://doi.org/10.1023/B:MACH.0000008084.60811.
 49
- [34] Hugo Thimonier, Fabrice Popineau, Arpad Rimmel, and Bich-Liên Doan. 2024. Beyond Individual Input for Deep Anomaly Detection on Tabular Data. In Proceedings of the 41st International Conference on Machine Learning, ICML 2024, 21-27 July 2024, Vienna, Austria (Proceedings of Machine Learning Research, Vol. 235). PMLR.
- [35] Hugo Thimonier, Fabrice Popineau, Arpad Rimmel, Bich-Liên Doan, and Fabrice Daniel. 2023. Comparative Evaluation of Anomaly Detection Methods for Fraud Detection in Online Credit Card Payments. arXiv:2312.13896 [cs.LG]
- [36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In Advances in Neural Information Processing Systems, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/ 3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- [37] Qi Wei, Yinhao Ren, Rui Hou, Bibo Shi, Joseph Y. Lo, and Lawrence Carin. 2018. Anomaly detection for medical images based on a one-class classification. In *Medical Imaging 2018: Computer-Aided Diagnosis*, Nicholas Petrick and Kensaku Mori (Eds.), Vol. 10575. International Society for Optics and Photonics, SPIE, 105751M. https://doi.org/10.1117/12.2293408
- [38] Julia Wolleb, Florentin Bieder, Robin Sandkühler, and Philippe C. Cattin. 2022. Diffusion Models for Medical Anomaly Detection. In Medical Image Computing and Computer Assisted Intervention – MICCAI 2022, Linwei Wang, Qi Dou, P. Thomas Fletcher, Stefanie Speidel, and Shuo Li (Eds.). Springer Nature Switzerland, Cham, 35–45.
- [39] Sun Yanmin, Andrew Wong, and Mohamed S. Kamel. 2011. Classification of imbalanced data: a review. International Journal of Pattern Recognition and Artificial Intelligence 23 (11 2011), 687–719. https://doi.org/10.1142/S0218001409007326
- [40] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes. In *International Conference on Learning Representations*. https://openreview.net/ forum?id=Syx4wnEtvH
- [41] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. 2016. Deep Structured Energy Based Models for Anomaly Detection. In Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48), Maria Florina Balcan and Kilian Q. Weinberger (Eds.). PMLR, New York, New York, USA, 1100–1109. https://proceedings.mlr.press/v48/zhai16.html
- [42] Hui Zhang, Zheng Wang, Zuxuan Wu, and Yu-Gang Jiang. 2023. DiffusionAD: Norm-guided One-step Denoising Diffusion for Anomaly Detection. arXiv:2303.08730 [cs.CV]
- [43] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. 2019. Lookahead Optimizer: k steps forward, 1 step back. In Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Cox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips. cc/paper_files/paper/2019/file/90fd4f88f588ae64038134f1eeaa023f-Paper.pdf
- [44] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. In *International Conference on Learning Representations*.

A DATASETS CHARACTERISTICS AND EXPERIMENTAL SETTINGS

A.1 Dataset characteristics

In Table 7, we display the main characteristics of the datasets involved in our experiments.

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 7: Datasets characteristics

B DETAILED EXPERIMENTS

B.1 Detailed tables for main experiments

Table 8: 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	Transformer	+KNN	+v-att.	+att-bsim	+att-bsim
					-bval
Wine	23.5±7.9	24.9 ± 5.9	26.5 ± 7.3	29.0 ±7.7	27.0 ± 5.6
Lympho	88.3±7.6	89.2±7.9	88.3±9.3	90.0±8.1	89.1±7.9
Glass	14.4±6.1	12.8 ± 3.9	12.2 ± 3.3	12.8 ± 3.9	11.1 ± 0.1
Verteb.	12.3 ± 5.2	15.7 ± 2.8	12.7 ± 4.1	14.3 ± 2.6	13.5 ± 5.1
Wbc	66.4±3.2	65.2 ± 4.0	66.2 ± 5.2	65.5 ± 4.4	67.9 ± 4.2
Ecoli	75.0 ± 9.9	75.6±7.5	75.6±9.7	73.8 ± 8.0	75.0 ± 9.7
Ionosph.	88.1±2.8	85.7 ± 3.4	$86.0 {\pm} 4.7$	91.7±2.1	79.3 ± 1.6
Arrhyth.	59.8 ± 2.2	60.3 ± 2.2	60.2 ± 2.7	61.2±2.1	61.1±2.8
Breastw	96.7 ± 0.3	96.7 ± 0.3	96.8±0.3	96.7±0.3	96.7±0.3
Pima	65.6 ± 2.0	64.7 ± 3.1	64.0 ± 3.3	64.3 ± 2.4	67.0±1.5
Vowels	28.7 ± 8.0	$40.0 {\pm} 10.0$	49.1±11.1	44.5 ± 10.5	58.0±11.2
Letter	41.5 ± 6.2	32.9 ± 11.8	41.8 ± 11.5	43.7 ± 10.3	28.5 ± 7.1
Cardio	68.8±2.8	65.6±3.6	$62.8 {\pm} 6.9$	67.7±3.7	68.3 ± 4.5
Seismic	19.1±5.7	17.4 ± 5.5	19.5 ± 6.3	16.7±5.5	17.5 ± 5.4
Musk	100 ±0.0	100±0.0	100±0.0	100±0.0	100±0.0
Speech	6.8±1.9	6.3±1.4	5.7 ± 1.7	5.9 ± 1.5	5.9 ± 1.7
Thyroid	55.5 ± 4.8	55.5 ± 4.9	56.0±5.9	55.8±6.3	55.3 ± 6.6
Abalone	42.5±7.8	42.5 ± 9.5	49.8 ± 6.5	53.0±5.7	43.2 ± 9.1
Optdig.	61.1±4.7	70.7±16.5	51.5 ± 7.6	62.6 ± 6.5	22.8±5
Satimage2	89.0 ± 4.1	$86.8 {\pm} 0.4$	90.7 ± 2.6	93.2±1.7	64.2 ± 7.2
Satellite	65.6±3.3	58.6 ± 2.9	57.3 ± 3.0	71.9±1.5	53.7 ± 3.3
Pendig.	35.4 ± 10.9	52.1±9.0	39.0 ± 14.5	53.4 ± 9.8	34.2 ± 12.2
Annthyr.	29.9±1.5	30.4±1.9	30.3±1.5	30.3±1.6	30.5 ±1.4
Mnist	56.7 ± 5.7	64.2 ± 3.7	61.7 ± 1.0	61.6±1.0	56.7±1.9
Mammo.	17.4 ± 2.2	17.3 ± 2.4	15.5 ± 2.5	17.2 ± 3.0	17.7±2.7
Shuttle	85.3±9.8	90.8 ± 2.9	67.7±13.7	87.8±3.7	95.6 ±1.8
Mullcr.	100 ± 0.0	100 ± 0.0	100 ± 0.0	100 ±0.0	100 ± 0.0
Forest	21.3±3.1	18.6 ± 4.6	21.0 ± 5.9	24.9 ± 6.5	11.1 ± 4.1
Camp.	47.0±1.9	48.5 ± 2.1	43.3 ± 2.3	49.7±1.2	49.1±1.1
Fraud	53.4 ± 4.4	56.4±2, 1	56.3 ± 2.1	57.1 ± 2.1	55.2 ± 1.8
Backd.	$85.8 {\pm} 0.6$	86.1 ± 0.6	85.2 ± 0.7	85.3 ± 0.6	82.4 ± 1.3
mean	56.2	56.1	55.7	58.6	53.9
mean std	4.4	4.6	5.0	3.7	3.7

Table 9: Anomaly detection AUROC(↑). V	We perform 5% T-test to test w	whether the differences betw	een the highest metrics for
each dataset are statistically significant.			

Method	Transformer	+KNN	+v-att.	+att-bsim	+att-bsim -bval
Wine	61.4±6.7	60.4±5.4	62.1±6.4	63.5±7.8	64.5±5.6
Lympho	$99.5 {\pm} 0.4$	99.6 ± 0.4	99.6±0.5	99.7±0.3	99.7±0.3
Glass	61.2 ± 7.0	61.2 ± 5.0	62.1±7.0	59.3 ± 6.9	59.1 ± 5.8
Vertebral	44.8 ± 5.2	46.7 ± 4.1	45.3±7.1	45.4±3.7	45.4 ± 4.7
WBC	95.0±1.1	94.3±1.5	94.3±1.6	94.2 ± 1.1	95.5±1.6
Ecoli	84.8±1.6	$84.8 {\pm} 1.8$	85.2 ± 2.7	87.4±1.8	85.4 ± 2.3
Ionosph.	95.4±1.9	93.7±2.7	$93.6 {\pm} 4.0$	97.5±0.1	87.2±2.3
Arrhyth.	81.7±1.1	81.9±0.9	81.8±0.9	82.3 ± 0.7	82.1±0.9
Breastw	99.6±0.1	99.6±0.1	99.6±0.1	99.6±0.1	99.6±0.1
Pima	67.2 ± 2.4	66.0±3.8	65.4±3.8	65.8 ± 2.9	68.7±1.4
Vowels	78.4 ± 9.2	86.1±5.2	90.4 ± 4.7	88.3±4.5	94.3 ± 2.8
Letter	80.5 ± 4.8	73.5 ± 9.6	81.0±8.7	81.5±6.8	69.1±7.7
Cardio	93.5±1.3	92.0 ± 1.7	89.9 ± 4.2	93.3±1.7	93.7±1.3
Seismic	58.2±7.9	56.8 ± 8.4	57.9 ± 7.6	58.0±6.7	54.8 ± 6.2
Musk	100±0.0	100 ± 0.0	100 ± 0.0	100 ±0.0	100 ± 0.0
Speech	47.2 ± 0.7	47.3 ± 0.8	47.3 ± 0.8	47.3 ± 0.8	47.0 ± 0.5
Thyroid	93.8 ± 1.2	93.8 ± 1.2	93.6±5.9	93.7±1.5	93.6±1.8
Abalone	88.3±2.0	86.1±3.6	88.0±3.5	87.9±3.7	86.6±2.9
Optdig.	96.4±4.7	96.2±9.8	94.9 ± 1.7	96.7±1.1	83.4±3.2
Satimage	$99.7 {\pm} 0.1$	99.5 ± 0.2	99.6 ± 0.2	99.8 ± 0.1	96.8 ± 1.9
Satellite	73.8 ± 2.5	68.9 ± 2.0	67.8 ± 2.5	79.5±1.9	62.0 ± 2.9
Pendigits	$93.8 {\pm} 2.6$	96.5 ± 1.4	94.1 ± 2.8	97.1±1.2	89.4±7.1
Annthyr.	$65.4{\pm}1.4$	66.0±1.7	66.2±1.3	66.2 ± 1.6	66.0 ± 1.1
Mnist	87.4 ± 3.2	90.3 ± 2.2	$89.9 {\pm} 0.4$	90.0±0.5	87.3 ± 1.0
Mammo.	77.6 ± 1.0	$76.8 {\pm} 2.4$	75.2 ± 2.9	78.4 ± 1.6	79.8 ± 1.8
Mullcross	100 ±0.0	100 ± 0.0	100 ± 0.0	100 ± 0.0	100 ± 0.0
Shuttle	97.2 ± 2.2	$98.1 {\pm} 0.5$	90.2 ± 6.1	97.7 ± 0.9	98.9±0.3
Forest	$95.1 {\pm} 0.8$	94.5 ± 0.7	$95.0 {\pm} 1.0$	95.4±0.9	92.7 ± 0.7
Campaign	75.3 ± 2.1	75.7±1.9	69.3 ± 2.0	76.1 ± 2.0	75.9 ± 2.1
Fraud	$94.7 {\pm} 0.4$	$95.1 {\pm} 0.4$	$95.2 {\pm} 0.4$	95.8 ± 0.4	94.7 ± 0.4
Backdoor	95.1 ± 0.2	95.2 ± 0.3	94.5 ± 0.2	94.7 ± 0.1	91.7 ± 0.3
mean	83.4	83.1	83.1	84.4	82.1
mean std	2.4	2.8	2.6	2.0	2.3

B.2 Synthetic dataset generation and experimental detail for section 5.1

The synthetic three dimensional dataset was generated as follows.

- Normal samples: We consider an interval of values [-2, 3] from which we uniformly sample the first feature x_1 . We then sample x_2, x_3 following the relation given in eq. (12) with parameters, $(\alpha_1, \beta_1) = (2, 3), (\alpha_2, \beta_2) = (4, 3)$ and $\varepsilon \sim \mathcal{N}(0, 1)$.
- Anomalies (type 1): We consider an interval of values [3.3, 4] from which we uniformly sample the first feature x_1 and keep the rest as for normal samples.
- Anomalies (type 2): We consider an interval of values [1.5, 2.5] from which we uniformly sample the first feature x_1 and sample x_2, x_3 following eq. (12) but with parameters (α_1, β_1) = (-7.5, -1) and (α_2, β_2) = (4, 3).

The vanilla transformer and its augmented version were trained with the following hyperparameters:

- Batch size: -1.
- Patience: 100 epochs.
- Learning rate (lr): 0.001.
- Hidden dim (*e*): 8.
- Masking probability *p_{mask}*: 0.15.
- Number of attention heads: 2.
- Number of layers of the encoder: 2.
- Retrieval hyper-parameters:
- Retrieval location: post-encoder
- Retrieval aggregation location: post-encoder
- $-\lambda: 0.5$
- $-C = D_{train}$

 $- card(\mathcal{H}) = 500$

For Mask-KNN, following [34] we set the number of neighbors to k = 5.