Wrocław University of Science and Technology
Faculty of Information and Communication Technology

Doctoral Thesis

Open-set classification
of high-dimensional data

Klasyfikacja z grupą otwartą
w danych wysokowymiarowych

Author:
MSc Eng Szymon Tomasz Datko

Supervisor:
PhD DSc Eng Henryk Maciejewski

Wrocław 2024

„We don't make mistakes;
we just have happy accidents."

– Bob Ross

Acknowledgments

I would like express my sincerest gratitude to my supervisor, Henryk Maciejewski, for the whole his help, not only during my research and redaction of this dissertation, but also, most importantly, for his assistance in introducing me the University ecosystem and guiding me how to survive in it. After all our countless conversations, I understand how to continue doing my part well, while still remaining a good and a kind person.

I would like to thank my fiancée, Sylwia, for invaluable support, unlimited patience and especially for believing in me and not asking about my daily writing progress ;-) I hope that now we will be able to fully focus on our future adventures together.

I would also like to send my thanks to everyone who made this whole period bearable and kept me sanity during this journey – most importantly: Adrian, Arie, Artur, Arx, Bogna, Daniel, David, Dawid, Ignacio, Karolina, Klaudia, Mateusz, Mike and Piotr. Without You, my motivation would have been lost long ago...

Finally, I would like to express my deep admiration and respect to Alexandra Asanovna Elbakyan, who has enabled free access to the scientific publications for countless number of researchers and scientists around the world, making the science open and not closed behind paywalls.

To all of You, from the deepest bottom of my heart,
Thank You so much! / Dziękuję bardzo! / Tack så mycket!

– Szymon Datko

Abstract

This work explores the topic of open-set classification problem in high-dimensional data. It is a task of identifying new data – outliers or out-of-distribution examples (OOD), that significantly differ from any previously available/known samples, i.e., training data used to build the closed-set classifier. While this task is strongly grounded and described with a solid statistical background in general, it turns out to be challenging and still not resolved for high-dimensional feature spaces, where all the current approaches and measures are proving to be insufficient. Furthermore, although many OOD detection methods for this task were proposed recently, the vast literature provides contradicting recommendations on solutions.

The motivation for undertaking this research problem lies in the rapid progress and astonishing performance of deep learning models (CNNs or ViTs) for images classification reported in benchmarks. These models involve high-dimensional feature representations ($d \sim 10^3$), however they are still based on closed-set recognition. Recent advancements in the machine learning domain and popularization of the artificial intelligence tools, such as the emergence of the complex deep learning techniques and the growing interest in self-driving cars, as well as other autonomous vehicles – all this makes the issues of reliability and safety extremely important topics nowadays. The problem of outliers detection fits into this theme, as one of the crucial aspects related to the machine learning models robustness is the models ability to adapt to the new data and situations. In any real-world and safety-critical implementation of machine learning-based systems, the reliable OOD-detection is a fundamental requirement to ensure safety in cases missed or not considered during the model training. Yet, as pointed out by leading scientists in the field, such aspects remain under-explored in literature so far.

This dissertation consists of three major parts. In first sections, the necessary background of outlier detection techniques is covered, focusing on distinguishing main approaches already proposed in literature. A detailed description of selected *post-hoc* methods is provided, as well as the required formalization and notation of the open-set

classification task. The main research interest is concentrated on methods that are implemented to work in the feature spaces because of their universality – the possibility to apply them for any existing, pre-trained model.

Then, the biggest chapter describes the results of a conducted numerical study on the simulated data distributions. The performance of selected *post-hoc* methods for outliers detection is analyzed, considering such factors as dimensions of feature vectors, numbers of training samples and distance to outliers – to examine how well various methods can distinguish both training and testing samples from the outliers. Additionally, the effects of correlations presence in the data on the methods performances are analyzed, as well as the behaviors of the methods when the features are characterized by non-uniform variances, i.e., data are unstandardized. The conducted research shows non-obvious behaviors of some of the examined methods, which is especially visible in higher dimensions of feature vectors. The work presents that the methods possess significantly different potentials for distinguishing between the known data (in-distribution, ID) and the unknown data (out-of-distribution, OOD). Moreover, the research identified the required conditions for methods to maintain accurate representations of data.

Finally, an evaluation involving the real-world data is performed. A wide range of pre-trained representation algorithms is used to obtain the feature vectors representations of text documents and image data, that are then examined for their potential in the open-set classification task with respect to the training data. A number of significant differences between the representations are observed and discussed. It turns out that the properties of representations greatly impact the performance of OOD detectors in the task. Hence, guidelines for selection of methods suitable for a particular representation can be formulated. It is shown that for all analyzed cases there exist a notable number of classes that contain samples much more difficult to distinguish from outliers, hence the per-class analysis of ID-OOD separability is proposed for the safety-critical applications. Such evaluation allows to identify security gaps and risks related to classes with poor OOD-generalization, that may require more in-depth analysis.

The conducted research contributes to the field by providing a novel insight of the selected *post-hoc* methods behaviors and properties in the high-dimensional feature spaces. The recommendations for the selection, usage and calibration of OOD methods for particular data representations in the outliers detection task are provided, involving applications on image and text data.

Keywords

- Open-Set Classification.
- Out-of-Distribution Detection.
- Measures of Outlierness.
- High-Dimensional Feature Vectors.
- Data Representation Techniques.

Streszczenie

W ninejszej pracy podjęto problem klasyfikacji z wykorzstaniem zbioru otwartego w danych wysokowymiarowych. Jest to zadanie polegające na wykrywaniu nowych danych – wartości odstających lub przykładów spoza rozkładu (ang. *out-of-distribution*, OOD), które znacząco różnią się od wszelkich wcześniej dostępnych/znanych próbek, czyli danych treningowych, wykorzystywanych do uczenia klasyfikatora zamkniętego. Chociaż zadanie to jest mocno ugruntowane i opisane na solidnym tle statystycznym w ogóle, okazuje się ono trudne i wciąż nierozwiązane w przypadku przestrzeni cech o wysokim wymiarze, gdzie wszystkie obecne podejścia i środki okazują się niewystarczające. Co więcej, chociaż zaproponowano wiele metod wykrywania danych odstających w takim zadaniu, to w obszernej literaturze można znaleźć sprzeczne rekomendacje na temat opisywanych rozwiązań.

Motywacją do podjęcia tego problemu badawczego jest znaczący postęp i zdumiewająca skuteczność modeli głębokiego uczenia się (CNN lub ViT) w zadaniach klasyfikacji obrazów, jakie raportowane są w analizach porównawczych. Stosowane modele obejmują wysokowymiarowe wektory cech ($d \sim 10^3$), jednak nadal opierają się na rozpoznawaniu w ramach zbiorów zamkniętych. Popularyzacja narzędzi sztucznej inteligencji i ostatnie postępy w dziedzinie uczenia maszynowego, takie jak pojawienie się złożonych technik głębokiego uczenia oraz rosnące zainteresowanie samochodami autonomicznymi, a także innymi pojazdami tego typu - wszystko to sprawia, że zagadnienia niezawodności i bezpieczeństwa są obecnie niezwykle ważnymi tematami. Problem wykrywania danych odstających wpisuje się w tę tematykę, ponieważ jednym z kluczowych aspektów związanych z niezawodnością modeli uczenia maszynowego jest zdolność modeli do adaptacji do nowych danych i sytuacji. W każdej rzeczywistej implementacji systemów opartych na uczeniu maszynowym, gdzie bezpieczeństwo ma charakter krytyczny, niezawodne wykrywanie OOD jest podstawowym wymogiem, zapewniającym stabilność działania w przypadkach przeoczonych lub pominiętych podczas uczenia modeli. Jednak, jak podkreślają czołowi naukowcy w tej dziedzinie, takie aspekty są jak dotąd ciągle niedostatecznie zbadane w literaturze.

Niniejsza rozprawa składa się z trzech zasadniczych części. W pierwszej części omówiono niezbędne tło dotyczące technik wykrywania danych odstających, koncentrując się na rozróżnieniu głównych podejść zaproponowanych już w literaturze. Podano szczegółowy opis wybranych metod *post-hoc*, a także wymagany formalizm i notację dla zadania klasyfikacji z uwzględnieniem zbioru otwartego. Główne zainteresowania badawcze skupiają się w pracy na metodach pracujących w przestrzeniach cech, ze względu na ich uniwersalność – możliwość zastosowania ich do dowolnego już istniejącego, wcześniej wytrenowanego modelu.

Następnie, w dominującym objętościowo rozdziale, opisano wyniki przeprowadzonych badań numerycznych na symulowanych rozkładach danych. Przeanalizowano skuteczność wybranych metod *post-hoc* w wykrywaniu danych odstających, biorąc pod uwagę takie czynniki, jak wymiary wektorów cech, liczba próbek uczących i odległość do przykładów odstających – aby sprawdzić, jak dobrze różne metody potrafią odróżnić zarówno próbki treningowe, jak i testowe, od danych odstających. Dodatkowo analizowany jest wpływ obecności korelacji w danych na działanie metod oraz zachowanie się metod, gdy cechy charakteryzują się niejednorodnymi wariancjami, czyli gdy dane są niezestandaryzowane. Przeprowadzone badania wykazują nieoczywiste zachowania niektórych z badanych metod, co jest szczególnie widoczne przy wysokich wymiarach wektorów cech. W pracy wykazano, że metody te posiadają znacząco różne możliwości rozróżnienia danych znanych (ang. *in-distribution*, ID) od danych nieznanych (ang. *out-of-distribution*, OOD). Ponadto w badaniu określono warunki wymagane, aby metody mogły zapewnić wierne odwzorowanie danych uczących.

Na koniec przeprowodzana jest konfrontacja metod z danymi ze świata rzeczywistego. Szeroki zakres wstępnie wytrenowanych algorytmów reprezentacji danych jest wykorzystywany do uzyskania wektorów cech z dokumentów tekstowych i danych obrazowych, które następnie są badane pod kątem ich potencjału w zadaniu klasyfikacji ze zbiorem otwartym w odniesieniu do danych uczących. Zaobserwowano i omówiono wiele znaczących różnic pomiędzy technikami reprezentacji. Okazuje się, że właściwości reprezentacji mają duży wpływ na skuteczność detektorów OOD w postawionym zadaniu. Można zatem sformułować wytyczne dotyczące wyboru metod odpowiednich dla konkretnej reprezentacji. Pokazano, że dla wszystkich analizowanych przypadków istnieje znaczna liczba klas zawierających próbki znacznie trudniejsze do odróżnienia od wartości odstających, dlatego też dla zastosowań krytycznych dla bezpieczeństwa proponuje się analizę separowalności ID-OOD dla poszczególnych klas. Taka ocena pozwala zidentyfikować luki bezpieczeństwa i ryzyka związane z klasami o słabej generalizacji w zadaniu wykrywania OOD; klasy te mogą wymagać dokładniejszych analiz w postawionym zadaniu.

Przeprowadzone badania stanowią wkład dziedzinę, dostarczając nowego wglądu w zachowania i właściwości wybranych metod *post-hoc* w odniesieniu do wysokowymiarowych przestrzeni cech. Podano zalecenia dotyczące wyboru, stosowania i kalibracji metod OOD dla poszczególnych reprezentacji danych w zadaniu wykrywania danych odstających, uwzględniając zastosowania w danych obrazowych i tekstowych.

Słowa kluczowe

– Klasyfikacja z wykorzystaniem zbioru otwartego.

– Wykrywanie danych spoza rozkładu.

– Miary odstania danych.

– Wysokowymiarowe wektory cech.

– Techniki reprezentacji danych.

Contents

Chapter 1

Introduction

1.1 Motivation – problem formulation

The open-set classification is a task focused on identifying new data, i.e., data previously unseen and not related to any originally known category, during the classification process, performed by a trained machine learning algorithm. Such task turns out to be important in any real-world implementation scenario [1], where samples from previously not considered classes can be fed into the system [27]. Therefore detection of such samples which are out-of-distribution (OOD) with regard to the training data of the machine learning model, becomes the essential element for safety-critical applications of ML [28]. However, while this task is already well established in literature [32][10] in general, it is still not only open, but also surprisingly highly under-explored when applied to the high-dimensional data, such as feature vectors or representations generated by deep learning models for image and text recognition. Although many OOD detection methods for deep learning models have been proposed recently [22], the literature provides contradicting conclusions and recommendations [65][83], as performance of OOD detection methods is highly benchmark-dependent.

Over the past decade, a numerous approaches for solving the problem were proposed in publications [43][26], aiming the performance improvements of deep learning models [79][62]. Yet, although in presented benchmarks the results seem indicating that the task is successfully solved [83], there are notable failures and mistakes observed in the real-world applications. While autonomous car reacting unexpectedly to a specifically painted t-shirt, or to a drawing on a billboard, may appear funny[1], in fact it indicates

[1] https://www.motortrend.com/news/can-a-t-shirt-stop-a-waymo-driverless-taxi-vehicle/

serious flaws in the complex machine learning-based systems that are more and more widely used nowadays. The same car may therefore miss an obstacle or even worse – a living person, leading to a seriously dangerous situation on the road.

The machine learning models are known to be affected by artificially introduced additional perturbations of inputs that may significantly change the models predictions [13] (adversarial attacks). Even a modification of a single image pixel can lead to a spurious response from a model [61] (one pixel attack). Moreover, a literature by Szyc et al. [63] questions the reliability and methodological standards of the benchmarks available in lately published papers, as it turns out that the outcomes may be drastically different for various parameters and starting conditions selected when the model is trained. Therefore, depending on the publication chosen, contradictory results may be observed, leading to conflicting recommendations.

Recent publication by leading scientists in the field, Bengio et al. [2], draws attention to the problem and points out the recommendations, criticizing the fact that the vast majority of recent work focuses on raw benchmark results and less on ensuring the stability of methods and examinations of the phenomena that are key to the security and safety of real-world implementations. The authors suggest that there is a need for reorienting the research and development works more towards the improved robustness, explainability and transparency of the models. This includes the models' abilities to respond to new situations and to detect unexpected, previously unseen data samples. Hence, the task of open-set classification/OOD detection remains not only an open research problem in deep learning, but it is also an important practical problem to be resolved in real-world, safety-critical applications of deep learning methods.

This dissertation contributes to the field of open-set classification/OOD detection by deeply studying the selected *post-hoc* OOD detection methods, implemented in high-dimensional feature spaces – providing new insight and increasing the understanding of the observed phenomena related to the flaws and strengths of different OOD detectors in high dimensional representations generated by different DL models. First, the numerical research on simulated data is conducted to identify the methods behaviors and properties in high dimensional data. Then, in the second part, the performance of OOD detectors (*post-hoc* methods, operating in the feature space) is analyzed in the task of image recognition, utilizing various deep learning architectures, e.g., convolutional networks (CNN), vision transformers (ViT) and models trained on pair of images with descriptions (CLIP). It is shown that the results depend on the properties and characteristics of the representations, i.e., feature vectors generated by various generating models. A number of measures and techniques are proposed to assess the risk of deep learning models implementations in real-world tasks due to the problem of

OOD-generalization, i.e., susceptibility to errors in recognizing outliers, as well as to better calibrate the operating point of OOD detection methods, i.e., rejection threshold. Similar analyzes are conducted for text representations as well.

1.2 Main contributions

The key contributions of this dissertation are as follows:

1. Conducted a comprehensive study on the performance of selected *post-hoc* OOD detection methods for outliers detection, measured more detail than in current literature (not only standard AUROC score, but also sensitivity and specificity), considering such factors as dimensions of feature vectors d, numbers of training samples n and distance to outliers h – to examine how well various methods can distinguish both training and testing samples from the outliers; the values of n and d parameters reflected the characteristics of the training data occurring in the popular OOD detection benchmarks for image data, e.g., deep learning models based on the ImageNet dataset (section 3.1).

2. Analyzed how the selected methods react to the presence of correlations in the data and identified group of methods that are performing well in such conditions and methods that are susceptible to increased errors in that case (section 3.2).

3. Verified how the methods perform under non-uniform variance of features, identifying methods that effectively consider that parameter and are usable for unstandardized data (section 3.3).

4. Conducted a research on the ability of outliers detection methods for obtaining accurate model of the training data, identifying the methods' requirements to achieve and maintain such accuracy (section 3.4).

5. Analyzed the errors in the estimation of the covariance matrix values, based on the high-dimensional training data – and identified the impact of these errors on the performance of OOD/outlier detection methods that rely on those estimations (sections 3.5, 2.3.6 and 2.3.7).

6. Proposed that the outliers detection techniques shall be compared not only by their ability to separate in-distribution and out-of-distribution data, but also on their classification performance (sensitivity, specificity) when calibrated on TPR = 95% with respect to the training data (section 2.2.4).

7. Identified that some of the popular SoTA (*State-of-The-Art*) methods may require calibration of threshold with the additional validation data for the effective realization of open-set classification in higher dimensions (section 3.1).

8. Proposed that OOD detection research shall go beyond showing the overall AUROC measure, the current literature standard, in favor of per-class analysis – presenting AUROC scores calculated per-class instead of a single average AUROC value provides additional insight into the safety risks of models' deployments due to classes with low OOD-generalization, which is especially important for safety-critical applications and shall be preferred in the OOD detection task benchmarks (section 4.3).

9. Identified that the performance of the outliers detection in real-world applications is tightly related to the representation-generating model used for the data processing – images or text documents; this phenomenon has not enough attention in the literature, results are often presented for a fixed representation (usually ResNet), attempting to form general conclusions that turn out not useful, as different representations favor different outliers detection techniques and can be recommended for a specific task configuration (sections 4.3 and 4.4).

10. Performed an analysis of the characteristics of the high-dimensional feature vectors produced from the various data representation-generating (deep learning) models for image and text recognition (section 4.5).

11. Shown that the popular data representation-generating models perform significantly different in terms of OOD-generalization. Hence, proposed a ranking of data representation models suitability for the outlier detection task (section 5.1).

12. Provided a Python library and a Python application for conducting a study of outlierness measures and performing an analysis of the experiments results (appendix B).

The author's existing contributions in the field are in addition as follows:

1. Performed a study of various outlierness measures proposed in literature, such as Angle-Based Outlier Factor (ABOF), Local Outlier Factor (LOF), k-Nearest Neighbors (kNN) and Mahalanobis distance, in the task of detecting abnormal data (outliers) in high-dimensional feature spaces. [71][78][15]

2. Proposed a generic two-step procedure for open-set classification of text documents represented by high-dimensional feature vectors. [71]

3. Proposed a new method to quantity the outlierness in high-dimensional data – IAOF (Interquartile Angle-based Outlier Factor). [71]

4. Analyzed the usability of feature vectors reduction techniques, such as Principal Component Analysis (PCA) and Random Projection (RP), when applied in the task of open-set classification. [76]

5. Conducted a study on different approaches to represent text documents with feature vectors, such as Bag of Words (BoW), Term Frequency – Inverse Document Frequency (TF-IDF), Word2Vec and fastText, in the task of subject classification of documents in Polish language. [72]

6. Studied the performance of Natural Language Processing (NLP) technologies compared to approaches not-requiring the language knowledge, depending on the number of training examples and feature vectors sizes. [73]

7. Proposed a solution for performing open-set classification of text documents, involving the fastText algorithm and the Local Outlier Factor measure. [75]

8. Analyzed how the dimensionality of feature vectors affects the classification performance for fastText algorithm and identified its ability to produce categories as focused projections in the feature-space. [77]

9. Studied how various distance metrics, such as euclidean and cosine distance, and the transformations of feature vectors, such as standardization and normalization, affect the results in the task of open-set classification of text documents. [74]

10. Studied the problem of precision-recall relation and finding the threshold value in the open-set classification task – as reduction of incorrect assignments comes with a risk of rejecting also correctly labeled data. [78]

11. Measured how the correlation structure in data (both correlation strength and the number of correlated variables), along with the dimensionality of feature vectors and the distance of outliers from typical data, affects the performance of outlierness measures. [14]

12. Conducted a research on the relation between the data representation models and the performance of outlierness measures in the task of outliers detection. [15]

1.3 Document organization

This document is organized as follows.

Chapter 2 covers the necessary background of outlier detection techniques, focusing on distinguishing main approaches already proposed in literature. A detailed description of selected *post-hoc* methods is provided, as well as the required formalization and notation of the open-set classification task. The techniques of representing real-world data as feature vectors are also described.

Chapter 3 presents the research conducted on simulated data that came from pseudorandom number generators (PRNGs). The performance of selected *post-hoc* methods for outliers detection is analyzed, considering such factors as dimensions of feature vectors, numbers of training samples and distance to outliers – to examine how well various methods can distinguish both training and testing samples from the outliers. Additionally, the effects of correlations presence in the data on the methods performances are analyzed, as well as the behaviors of the methods when the features are characterized by non-uniform variances, i.e., data are unstandardized.

Then, chapter 4 contains the results of research run on the real-world data. A wide range of pre-trained representation algorithms is used to obtain the feature vectors representations of text documents and image data, that are then examined for their potential in the open-set classification task with respect to the training data. A number of significant differences between the representations are observed and discussed.

Finally, chapter 5 is the summary of the work, discussing the conclusions, impact on the field and potential future research to be conducted. Recommendations for improving OOD detection in high-dimensional data are provided.

Chapter 2

Related work

2.1 Open-set classification

Traditional, closed-set classification algorithm is designed to recognize data and assign a label corresponding to one of the classes (categories) known during the model training. Such algorithm can perform exceptionally well, achieving superhuman accuracy in the task, as long as the input data truly correspond to the trained categories. However, when such model is exposed to new, unexpected, previously unseen data categories, it still assigns one of the originally known labels, resulting in incorrect outcome.

The open-set classification aims to resolve the problem by extending the model's capabilities with additional output category: open-set, indicating that the analyzed input data are not similar enough to any previously known data class. Such ability is especially important for any practical scenarios of Machine Learning models applications, as in real-world the new data points can emerge constantly. This task is commonly known in literature also as the outlier detection, the open-world recognition and the out-of-distribution detection. Many methods were proposed in literature to solve this problem, summarized in surveys [32][10].

Recently, with huge success of Deep Learning models for image and text recognition tasks, a dedicated line of literature appeared to resolve the problem of open-set classification / out-of-distribution detection in high-dimensional representations generated by such models. Many methods were proposed as state-of-the-art algorithms: Mahalanobis distance with pooled covariance matrix [43], k-Nearest Neighbors [62], Integrated Rank-Weighted Depth [12], Energy-based OOD detector [45], training with outlier exposure [29], Virtual-logit Matching [79] and many others, summarized and compared in recent comprehensive surveys [22][83].

However, although multiple solutions were proposed, the problem remains unresolved, as there is no clear recommendation as to which method should be used in a practical application – and the benchmarks present contradicting results [65][83]. Additionally, the OOD detection in high-dimensional data is highly unstable, as shown in recent publication [63]. Hence, this dissertation is focused on exploring the problem of out-of-distribution detection in high-dimensional feature spaces, analyzing performance and properties of OOD detectors, utilizing high-dimensional representations generated by different Deep Learning models. The goal is to provide new insights and recommendations for reliable OOD detection in Deep Learning, which is crucial for safety-critical deployments of AI in real-world.

2.1.1 Outlier detection methods

There are three main approaches to the problem of out-of-distribution data recognition commonly recognized in the current literature [83]: *post-hoc* methods, training-time regularization and training with outlier exposure.

The ***post-hoc* methods** are a group of techniques that work on the outputs of existing models, e.g., logit layers fed to softmax function [7], or that involves the feature-space-wise analysis of the data vectors produced by models (i.e., penultimate layers) to identify the abnormal data. The detection of out-of-distribution data is typically realized with distance-like functions, such as the Mahalanobis distance [47] or Local Outlier Factor [6]. Section 2.3 contains the detailed description of selected methods. The significant advantage of *post-hoc* methods is their ease of use and universality, as they are model agnostic – the computations are performed in the feature-space and they require no additional modifications of the models that produce the feature vectors, so it is possible to utilize any existing and pre-trained model, making them applicable for all kind of real-world implementations (images, texts, ...). This is especially important nowadays, as the re-training of huge deep-learning models can be too expensive or even impossible for practical solutions – hence the *post-hoc* methods offer an effective solution. Therefore, the *post-hoc* methods that work in the feature-space are the main interest of this dissertation.

The alternative approaches are focused on improving the models' ability to detect outliers by either changing the models architecture and/or modifying the training process, which is usually achieved by adjusting the loss function (objective function). The **training with outlier exposure** assumes that a large collection of selected outliers is presented to the model that is then trained to minimize the outputs for such examples [29], i.e., effectively not assign any known label the outliers examples.

The problem with practical application of such methods is related to impossibility of obtaining dataset of example outliers that can stand for all possible unknown data that may appear in the future. Therefore, in contrast, the **training-time regularization** algorithms do not require additional outliers examples, instead utilizing techniques such as the contrastive learning [64] to generate class representations that are distant from each other. However, the biggest disadvantage of both mentioned approaches is their computational complexity and requirement of preparing (training) the model customized to a designated scenario, which is difficult and impractical with big deep-learning state of the art models available nowadays.

2.1.2 Near OOD vs Far OOD

The current literature distinguishes two major subtasks of outliers detection problem – identifying the near OOD data and the far OOD [81]. This distinguishing highlights the concept that in the real-world applications there are data points coming that show various degrees of similarity/differences from the previously known training samples. Intuitively, the distinction between animals such as dogs and wolves is much harder than distinction between animals and plants. While recognition of far OOD examples is considered much easier, dealing with the near OOD samples is still "a major challenge" [21] to be resolved. The near OOD samples are more prone to spurious assignments.

It shall be noticed that, contrary to semantic differences, what is near and what is far for a human intuition, it may have an entirely different meaning for the machine learning algorithm, i.e., in terms of the feature vectors space distances – because it all depends on the features identified and correlated with classes by a chosen representation model during the training process. In the effect, such non-obvious relations as presented in [57] can be observed, i.e., wolf class may be correlated with the presence of a snow in the training data.

2.2 Task formalization

2.2.1 Definitions and notation

The classification is a task of assigning elements, such as images, documents, etc., to a named group or category identifying elements that share the same properties, e.g., subject, authorship, etc.

The elements are represented using feature vectors, usually noted as v or x. Each feature vector is a d-dimensional array of real numbers, i.e., $v \in \mathbb{R}^d$, $x \in \mathbb{R}^d$. Each such real number – vector's component, e.g., marked v_j for j-th component, denotes the presence or strength of some attribute, so called feature, in the element, e.g., number of some word occurrences in a document or appearance of shape in an image – although the interpretation of individual components of feature vectors may be far less obvious, or even not doable at all, for some representation algorithms (section 2.4). The d is a dimension of the feature vector, $d \in \mathbb{N}$, equal to the number of vector's components. Feature vectors are subject to operations described by a branch of mathematics known as vector algebra.

A set of feature vectors x_i is called a cluster, $x_i \in K$. The n is the number of samples (feature vectors) in the cluster. Cluster K is often represented as a matrix of size $n \times d$, i.e., n rows and d columns. Hence, in that representation $K[i, j]$ (or $K_{i,j}$) denotes the j-th component of the i-th feature vector; also $K[i, *] = x_i$.

The group of elements that share the same properties is usually referred to as the class or the category. It is a label, denoted as c, typically a string or a number, that is assigned by a classifier to an element. The set of all classes, $c_l \in C$, includes all possible outcomes of a classifier, $|C| = m$. If there is a known class c_K pre–associated with all elements of a cluster K, then such cluster is often called a training cluster; a collection of training clusters for all classes $c_l \in C$ is called a dataset $\mathcal{D}$.

The dataset is typically represented as an augmented matrix that combines the matrix of all available feature vectors, $X = \left[K_1^\top | K_2^\top | ... | K_m^\top\right]^\top$, with a column vector of class labels y that associates each feature vector x_i with a class c_i, i.e., $\mathcal{D} = (X|y) \sim (x_i|c_i)$.

The classifier is an algorithm that assigns a class label to a given feature vector,

$$f(v) : \mathbb{R}^d \to C. \tag{2.1}$$

It may utilize any of machine learning techniques, such as neural networks, decision trees, support vector machines or probabilistic models, capable of distinguishing between data using a defined set of parameters. The function f is selected and designed to best fit the training data, e.g., based on the empirical risk minimization or the structural risk minimization [68], reducing the loss function related to $f(x_i) \neq y_i$ error. The process of identifying relevant parameters and their weighted importance related to features, utilizing a dataset with one or more training clusters, is referred to as the training of a model [24].

Summarizing, in short:

- v, x_i – a feature vector.
- K – a cluster, often represented as a matrix of feature vectors $x_i \in K$.
- n – a number of samples (feature vectors) in a cluster.
- d – a dimension of each feature vector.
- c – a class/category, i.e., a label assigned by a classifier, $c \in C$.
- m – a number of known classes, $m = |C|$.
- f – a classification algorithm, that assigns c for a given v.

2.2.2 Procedure for open-set classification

In this research, the *post hoc* approach for open-set classification in considered, involving two-step procedure described by Walkowiak et al. [71]. Assuming that for a given class c there exist a training cluster T and the task is to classify element v, it can be summarized as follows:

1. First, the traditional classifier performs a closed-set classification, assigning best possible candidate class c for a given feature vector v.
2. Second, the verification is made with respect to the available training samples, i.e. with respect to the known in-distribution (ID) data for the class c – rejecting the assignment if v is not similar to examples from cluster T.

The verification is a second classification task to perform, that involves a function OF to measure the (dis)similarity of the element v compared to the training examples from available dataset T – expressed as the score s. Several ideas and illustrations of such OF measure are discussed in section 2.3,

$$s = OF(v, T). \tag{2.2}$$

The decision function in that case compares the obtained score value s with a defined threshold value t, i.e., when s is greater than t we reject the assignment and classify v as the outlier,

$$f(v) = \begin{cases} c & \text{if } s \leq t \quad \Rightarrow \text{ID}, \\ \varnothing & \text{otherwise} \quad \Rightarrow \text{OOD}. \end{cases} \tag{2.3}$$

The threshold value t in the conducted research is selected *a priori* as the n-th percentile (P_n) of scores calculated for all elements within the cluster T – i.e., a value that is greater than $n\%$ of typical scores observed within that cluster,

$$t = P_n\Big(\big\{\ \forall v \in T:\ OF(v,T)\ \big\}\Big). \tag{2.4}$$

The value of the t can also be defined according to the commonly used standard statistical procedures to detect extreme observations in univariate distributions, e.g., based on the interquartile range (IQR) cutoff (e.g., proposed in [67][71]).

2.2.3 Verification of OOD detection in feature space

For evaluation described in chapters 3 (sections 3.1, 3.2 and 3.3) and 4 (sections 4.3 and 4.4), apart from the training data cluster T, two additional data clusters are utilized: the testing cluster K, representing known data that come from the same in-distribution as T; and the outliers data cluster U, containing examples that should not be assigned to any known class $c \in C$, i.e., out-of-distribution.

Each vector $v \in (K \cup U)$ is classified as either ID or OOD with respect to the training data T. The positive reply from the classifier is associated with a recognition of in-distribution data; the negative response from the classifier corresponds to detecting an outlier. The following four outcomes of classifier are possible:

- **TP** – **True Positive** – known data was correctly labeled as an inlier,
- **FN** – **False Negative** – known data was incorrectly labeled as an outlier,
- **TN** – **True Negative** – unknown data was correctly labeled as an outlier,
- **FP** – **False Positive** – unknown data was incorrectly labeled as an inlier.

Based on outcomes, results of classification can be measured with traditional metrics:

- **sensitivity** – the proportion of correct positive responses out of all expected,

$$sensitivity = \frac{TP}{TP + FN}\ , \tag{2.5}$$

- **specificity** – the proportion of correct negative replies out of all expected ones,

$$specificity = \frac{TN}{FP + TN}\ , \tag{2.6}$$

- **accuracy** – the overall proportion of correct predictions out of all assignments,

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \ . \tag{2.7}$$

The sensitivity is often referred to as True Positive Rate (TPR), while the specificity is also known as True Negative Ratio (TNR). The intuitive interpretation of those metrics in the context of conducted experiments is that the sensitivity describes the ability of correctly recognizing the in-distribution data, while the specificity is related to detector's capability of properly identifying out-of-distribution samples. The ideal, desired scenario is that there are no incorrect assignments made by the classifier, i.e., $FP = FN = 0$, hence $sensitivity = specificity = accuracy = 1.0$ ideally.

2.2.4 Calibration with respect to the training data

The literature standard is to measure and compare the OOD detectors performance by the AUROC scores – Area Under the Receiver Operating Characteristic. In this approach, the decision function values (score s, formula 2.2) are calculated for all known testing samples (ID) and available out-of-distribution examples (OOD), ignoring the relation to training data used to produce the machine learning model. Then, obtained values are sorted and iterated through, calculating True Positive Rate (TPR) and False Positive Rate (FPR) for each score value (treated as threshold here), i.e., counting the correctly classified in-distribution data and incorrectly classified out-of-distribution samples. This allows to produce the Receiver Operating Characteristic (ROC) curve and the area under it (AU-) can be computed numerically – summarized as $AUROC$, with ideal value being $AUROC = 1.0$; any value $AUROC \leq 0.5$ means the classifier performs worse than randomly given assignments.

Effectively, this measures the detectors abilities to correctly distinguish testing samples (ID) from outliers (OOD). However, as show the results of conducted research (chapter 3), even though some methods may promise effective distinguishing (i.e., reach high AUROC values), they also render testing samples as distant from the available training data. In some cases the testing samples may appear closer to the outliers than the actual training samples, although both testing and training samples come from the same distribution (figure 3.6). While in such cases the successful utilization of outlier detector is theoretically possible, it would require calibration according to the additional validation data, rather than with respect to the available training samples, which is vague and difficult to justify practically.

The existing literature [43] suggests to measure the performance of OOD detectors by calculating TNR value at $95\%TPR$. However the calculation is performed on testing in-distribution data, while the reliable calibration for real-world applications requires relying on training in-distribution samples only.

Hence, the contribution and proposal of this work is that the outlier/OOD detectors shall be additionally compared by the OOD detection performance when the OOD detection threshold is calibrated based on the ID training data (i.e., the classification task with respect to the training samples), not only by their ID-OOD separability potential (AUROC score) utilizing the testing data.

The additional criteria for evaluating OOD detection methods are then:

- sensitivity, measured as the fraction of correctly classified testing samples (ID),
- specificity, defined as the proportion of correctly recognized outliers (OOD),

when the OOD detection threshold t is selected at $95\%TPR$ of the training data, so that at least 95% of training data must be correctly classified by the model. (!)

The conducted study shows that the popular OOD detectors differ significantly in terms of the proposed measures (sections 3.1.2, 3.1.3 and 4.4). In particular, the Mahalanobis Distance and k-Nearest Neighbors reach very low sensitivity in the range of parameters (number of training samples n, dimension of feature vectors d) typical for the deep learning models (chapter 4). This bears a profound impact on the way the OOD methods should be calibrated in deployments supporting the real-world applications - as suggested in section 5.1.

2.3 Measures of outlierness used by OOD detectors

For quantifying the similarity of a given feature vector $v \in \mathbb{R}^d$ with respect to a specified candidate class c_T, represented by a training cluster T, we define a measure function $OF(v, T)$ such as that

$$OF(v, T) : \mathbb{R}^d \to \mathbb{R}, \tag{2.8}$$

i.e., for a given feature vector v of dimension d it assigns a single real number value as an output. That value can be intuitively interpreted as a distance of a given feature vector v from the cluster T, hence typically the greater value the more distant the vector

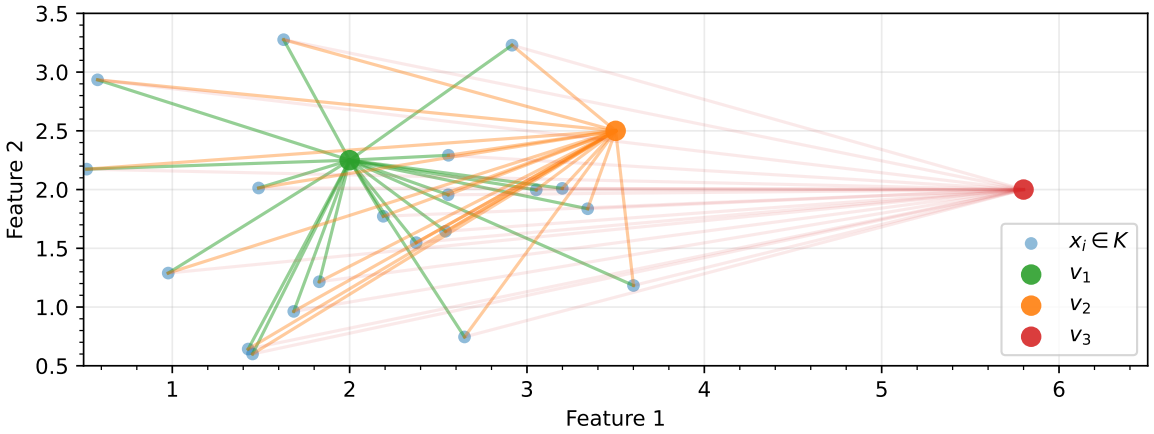


Figure 2.1: Idea of the Angle-Based Outlier Factor applied as an outlierness measure. Element v_1 is located inside the cluster T, element v_2 is on the edge of the cluster T and element v_3 is a distant outlier; lines drawn correspond to vectors involved in the outlierness score calculation.

is, although this is algorithm-specific and there are known algorithms that present the opposite behavior. The exact formula on how to calculate the output value is also specific to a particular algorithm – the differences include such factors as:

- how is the target cluster T being represented?
- which metric is used to calculate the distance?

It should be noticed that the outlierness measures usually do not define absolutely at which output value the given data vector v shall be marked as an outlier. Hence, the proposed procedure of finding a threshold value and performing the open-set classification, described in section 2.2.2.

2.3.1 Angle-Based Outlier Factor

The Angle-Based Outlier Factor (ABOF/ABOD), proposed by Kriegel et al. [40], is an anomaly detection technique that relies on the analysis of angles between feature vectors to determine whether the data is an outlier or not. Because of relying primarily on angles, instead of e.g., Euclidean distances, it is claimed to be effective even for applications in high-dimensional spaces.

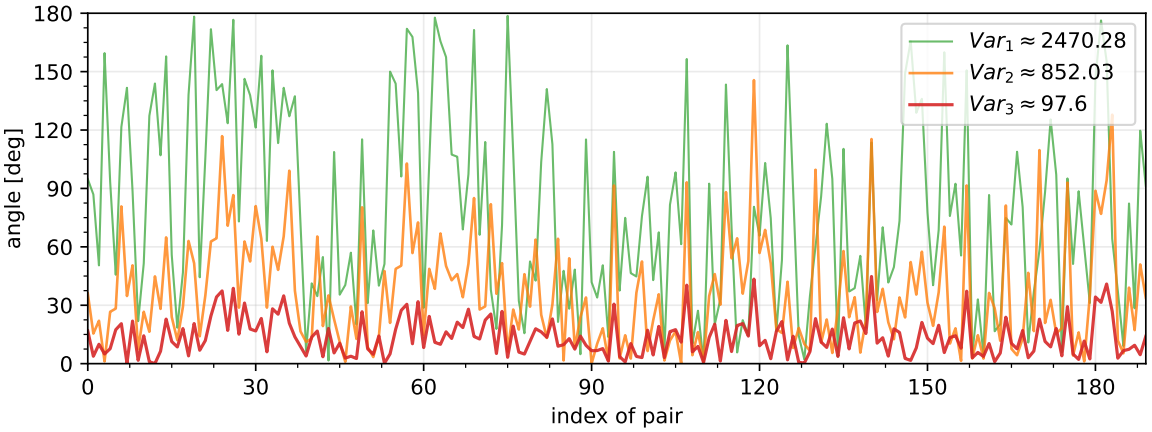


Figure 2.2: Ranges of angles between vectors observed for various examined examples (typical point – v_1, edge point – v_2, outlier – v_3). For 20 elements in cluster T, there are 190 unique pairs in total, so 190 angles possible for each of points v_1, v_2 and v_3. Highest variance is observed for an inlier, lowest variance in case of an outlier.

Figure 2.1 illustrates the intuitive idea behind the algorithm. For any data point v in the feature space $\mathbb{R}^d$ and a given data cluster T, there can be two vectors constructed between the point v and points x_1, x_2 randomly selected from the cluster T. Then, the angle α (or the value of $\cos(\alpha)$) between the constructed vectors may be calculated.

If the point v is located within the data cluster T, then a wide spectrum of angles values may be observed, i.e., both acute and obtuse angles, as presented in figure 2.2. Contrary, when the point v is an outlier, a dominant number of acute angles with small variability of values shall be observed.

Mathematically this can be quantified by calculating the variance of angles values between the given data point v and all possible pairs of points from the target cluster T. If the variance is high, i.e., a wide spectrum of angles is observed, the given data point is not an outlier. If the variance is low, the data point is likely to be an outlier.

Let P be the set of all unique pairs (x_1, x_2) – combinations of elements from the target cluster T. Then the outlierness score for a given vector v can be calculated as

$$ABOF(v,T) = \mathrm{Var}\left\{ \forall (x_1,x_2) \in P : \frac{\overrightarrow{x_1 - v} \cdot \overrightarrow{x_2 - v}}{\left\|\overrightarrow{x_1 - v}\right\|^2 \cdot \left\|\overrightarrow{x_2 - v}\right\|^2} \right\}. \tag{2.9}$$

It shall be noticed that because all unique pairs from cluster T are being considered, the computational complexity of the original algorithm is like $\mathcal{O}(n^3)$, hence for large datasets it rapidly becomes time ineffective. Therefore, even the original paper [40]

additionally proposes variants of ABOF: approxABOF and LB-ABOF, that are faster to compute, based on various approximations, e.g., by subsampling the cluster T to consider only T nearest neighbors of point v, at the risk of lower accuracy.

In addition it can be observed that in equation 2.9 the function formula is not based solely on the angular measure, because considering the scalar/dot product definition,

$$\vec{v_1} \cdot \vec{v_2} = \|\vec{v_1}\| \cdot \|\vec{v_2}\| \cdot \cos(\alpha), \tag{2.10}$$

the original formula 2.9 may be therefore presented in a following alternative form,

$$ABOF(v,T) = \mathrm{Var}\left\{ \forall (x_1, x_2) \in P : \frac{\cos\left(\angle(\overrightarrow{x_1 - v}, \overrightarrow{x_2 - v})\right)}{\left\|\overrightarrow{x_1 - v}\right\| \cdot \left\|\overrightarrow{x_2 - v}\right\|} \right\}, \tag{2.11}$$

where $\angle(\overrightarrow{x_1 - v}, \overrightarrow{x_2 - v})$ denotes the angle between vectors $\overrightarrow{x_1 - v}$ and $\overrightarrow{x_2 - v}$. This makes it visible that the original algorithm actually takes into account the angles that are normalized by the product of the length of the difference vectors [40]. When the analyzed point v is far from the cluster T, the calculated angles are therefore contributing less to the outcome score value. Hence, literature [71] also discusses the purely angle-based variants without such additional scaling, that turns more accurate in some cases,

$$ABOF2(v,T) = \mathrm{Var}\left\{ \forall (x_1, x_2) \in P : \cos\left(\angle(\overrightarrow{x_1 - v}, \overrightarrow{x_2 - v})\right) \right\}. \tag{2.12}$$

During the research the own implementation (Section B.2) was used, based on the description from the original article [40]. For convenience, the returned values were inverted, so the greater values indicate that data more likely to be outliers.

2.3.2 Euclidean distance

The Euclidean distance is an intuitive baseline method for quantifying the similarity based on the distance. In this approach, the data cluster T is represented by a middle point μ_T, calculated as an average of all ($n_T = |T|$) feature vectors x_i, $x_i \in T$,

$$\mu_T = \frac{1}{n_T} \cdot \sum_{i=1}^{n_T} x_i \ . \tag{2.13}$$

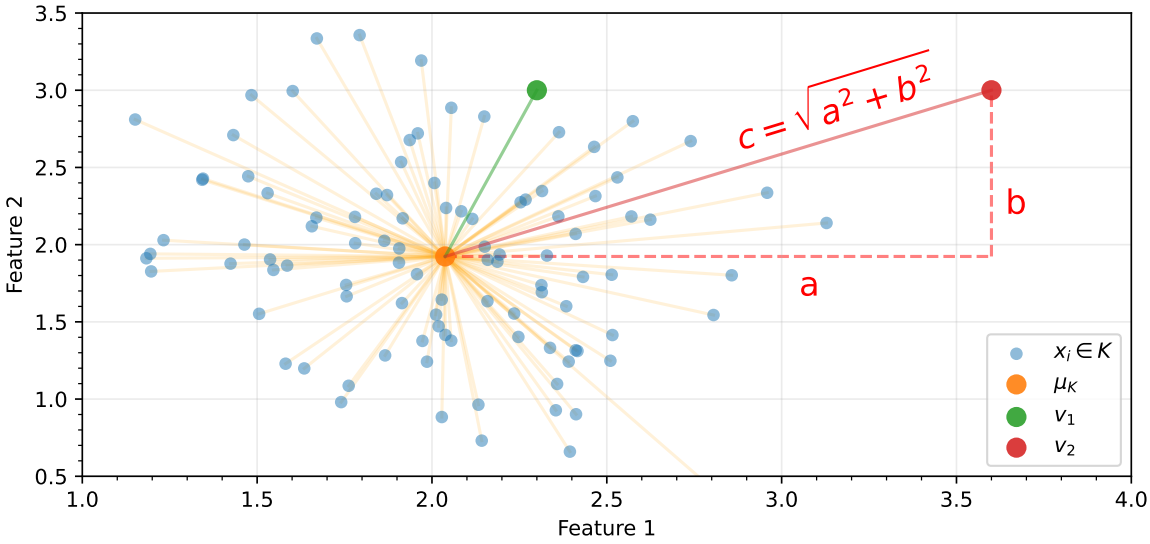


Figure 2.3: Idea of the Euclidean distance applied as an outlierness measure. The location μ_T of the cluster T center is identified and then involved in calculation of the outlierness scores (Minkowski metric of order 2) for elements v_1 and v_2.

The outlierness score for a given vector v against the data cluster T is then calculated as the Minkowski distance of order 2 in $\mathbb{R}^d$ space between the v and point μ_T location,

$$ED(v, T) = \left\| \overrightarrow{v - \mu_T} \right\|. \tag{2.14}$$

Figure 2.3 presents the idea of leveraging such simple method for outlier detection. Any element v that is further than acceptable threshold can be marked as outlier (v_2), wheres object at typical distance is considered as in-distribution (v_1). This approach is suitable as long as the data are evenly distributed on all axes in $\mathbb{R}^d$ space and also not affected by any correlation. Otherwise the risk of spurious ID/OOD label assignment increases, especially in high-dimensional feature spaces.

During the research the implementation from the SciPy library [70] was utilized.

2.3.3 Integrated Rank Weighted Depth

The Integrated Rank Weighted Depth, published by Ramsay et al. [54], is another method of quantifying the outlierness score. It utilizes the Monte Carlo-like approach and considers angles between feature vectors for producing a representation of the known/training set, with the similarity then determined using so called depth of data.

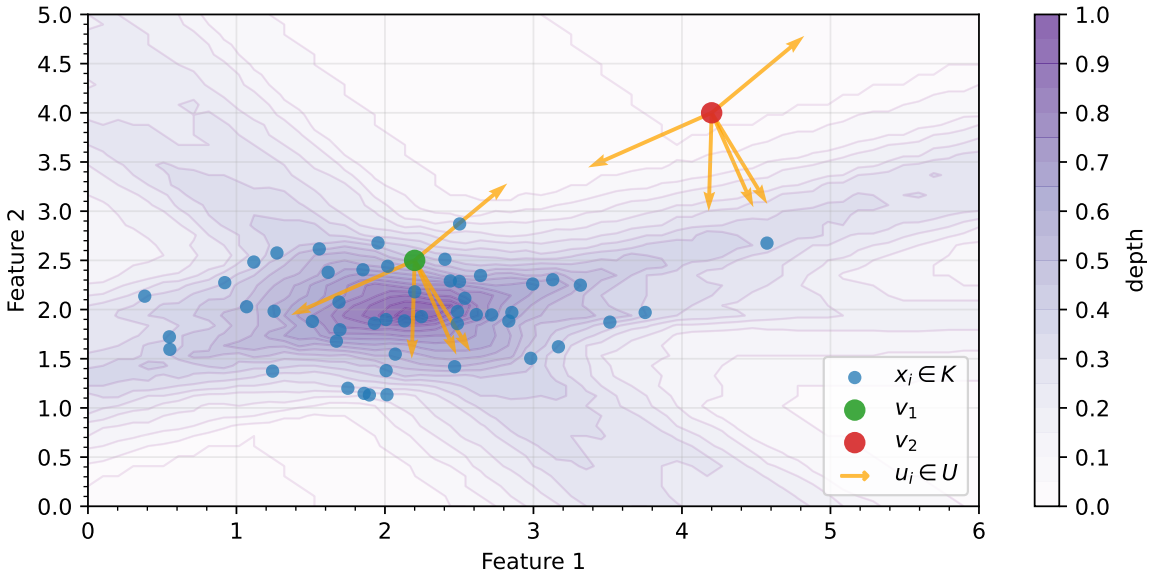


Figure 2.4: Idea of the Integrated Rank Weighted Depth applied as an outlierness measure. The contour plot is used to visualize the depths calculated for cluster T. Point v_1 is located in the region surrounded by cluster T points (high depth), while point v_2 is an outlier (low depth value). The projection vectors u_i involved in depth calculation are drawn for reference. The depth values are normalized for convenience.

First, a collection U of n_{proj} number of projection vectors u_j is created by randomly selecting directions from a unit hyper-sphere in $\mathbb{R}^d$ space. Then, a representation of known data cluster T is prepared by computing matrix M of size $n_T \times n_{proj}$, containing the scalar products between each feature vector $x_i \in T$ and projection vector $\overrightarrow{u_j} \in U$,

$$M[i,j] = x_i \cdot u_j \ . \tag{2.15}$$

Each i-th row in matrix M corresponds to the feature vector x_i of cluster T, while each j-th column corresponds to the scalar products values obtained for the projection vector u_j. Hence, there are n_T rows and n_{proj} columns; the $M[i,j]$ component is a value of the scalar product between vectors x_i and u_j.

To calculate the outlierness score for a given vector v against the data cluster T, the next step to compute the values of scalar products between v and the projection vectors $u_j \in U$, then subtracting the resulting values from the representation matrix M,

$$M_v[i,j] = M[i,j] - v \cdot u_j \ . \tag{2.16}$$

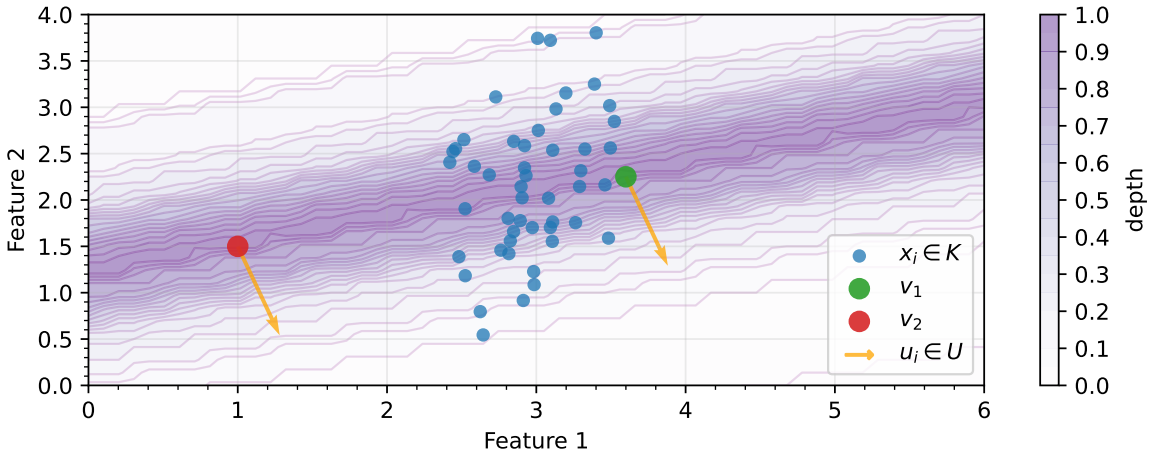


Figure 2.5: The IRWD algorithm identifying spurious correlation due to $n_{proj} < d$. In this case, element v_2 is considered as close to the cluster T as the element v_1. Utilizing more projection vectors u_i would help mitigating the problem.

The resulting matrix M_v is processed column-wise, which corresponds to analyzing results obtained for each projection vector u_j. For each column, the number of positive and negative values is counted and the lower number is selected. Finally, the outlierness score is calculated as a sum of selected numbers, divided by the number of M_v elements,

$$IRWD(v,T) = \frac{1}{n_T} \cdot \frac{1}{n_{proj}} \cdot \sum_{j=1}^{n_{proj}} \min \Big\{ \text{count}\{M[*,j] \leq 0\} \ , \ \text{count}\{M[*,j] > 0\} \Big\} \ . \tag{2.17}$$

Intuitively, the values of M_v matrix correspond to the distances and locations of each feature vector x_i in relation to the given v while looking along the direction u_j. In other words, when considering a plane defined by a point v and a vector u_j, the positive value of $M_v[i,j]$ means that given x_i is in front/above that plane, while the negative value means the x_i is behind such plane. Note that the actual individual distance itself is not relevant, there matters only the location that is linked to the sign (positive/negative) of the distance value. The score gets maximized, if there is an equal number of elements x_i on both sides of the plane. Hence, the idea of the data depth emerges as a way of characterizing and localizing the center of a distribution, which is promised to be suitable especially for non-Gaussian distributions, where only the traditional mean and median may not provide reliable understanding of data spread.

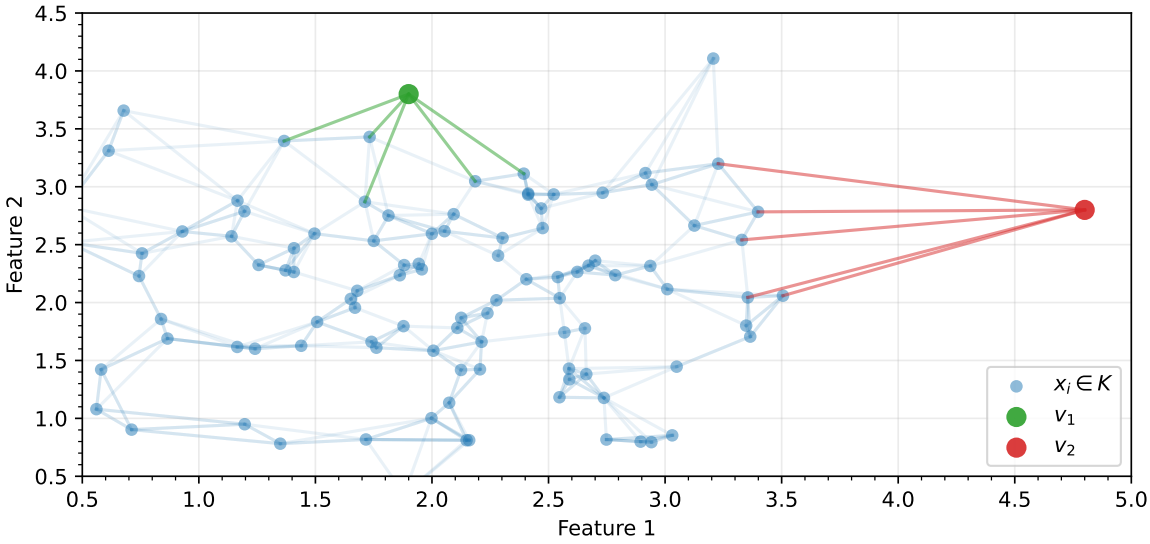


Figure 2.6: Idea of the k-Nearest Neighbors applied as an outlierness measure. The lines drawn connects elements to their $k = 5$ closest neighbors. Element v_1 is located close to the cluster T, so average distance to it's closest neighbors is lower than for element v_2 that is located farther.

The high depth value indicates that a data point v lies within the central region of the data cluster T, surrounded by neighbors. Contrary, low depth value suggests that a point is located on the outskirts of the data distribution, potentially indicating that the v may by an outlier. Figure 2.4 illustrates both such cases with the random projection vectors u_i drawn for the reference as well.

It is worth to notice that the chosen number of random projection vectors, n_{proj}, impacts the outcome of IRWD significantly. Especially, with n_{proj} lower than the dimension of $\mathbb{R}^d$ space, i.e., $n_{proj} < d$, there is a risk of identifying spurious correlations in data. Example of the such case was presented in the figure 2.5, where both in-distribution data (v_1) and the out-of-distribution sample (v_2) are ranked with the same depth score.

During the research the own implementation (Section B.2) was used, based on the description from the the NeurIPS publication [12]. For convenience, the returned values were inverted, so the greater values indicate that data more likely to be outliers.

2.3.4 k-Nearest Neighbors

The k-Nearest Neighbors is a well defined, fundamental algorithm of the machine learning [24], that is useful in variety of tasks, such as in clusterization and classification. It was recently proposed as a state-of-the-art out-of-distribution detector [62].

The core concept assumes the identification of given k number of elements from a known set T that are closest neighbors of a given element v. Such identification can be performed by using one from a number of possible underlying algorithms, for example a naive brute-force search or by utilizing sophisticated data structures, like with k-dimensional tree [3][8] or ball tree algorithm [50][44]. Also, for distance calculation, there may be any of the known metrics involved, such as the Manhattan/L1 distance or the general Minkowski metric; commonly the standard Euclidean distance is used [60].

In applications for outlier detection, there are two main approaches proposed in the literature. First solution relies on selecting the distance to the k-th neighbor as the outlierness score, i.e., the highest distance among all k neighbors around given v,

$$kNN_I(v,T) = \max\left\{ \forall x_i \in N_k(v,T) : \left\|\overrightarrow{v - x_i}\right\| \right\}, \tag{2.18}$$

where $N_k(v,T)$ is a function that returns the set containing k number of elements x_i from cluster T that are closest to v. The second approach relies on calculating the average distance to all of k considered neighbors,

$$kNN_{II}(v,T) = \frac{1}{k} \cdot \left(\sum_{x_i \in N_k(v,T)} \left\|\overrightarrow{v - x_i}\right\| \right). \tag{2.19}$$

Typically, the high values of the calculated outlierness score, above certain threshold, indicate that given feature vector v is far from all samples present in training cluster T, i.e., it is an outlier. Respectively, low values of the score indicate that the v is not distant from known examples, hence it is an inlier.

Figure 2.6 illustrates the representation of example cluster T from the kNN algorithm's perspective, along with in-distribution sample (v_1) and outlier (v_2). The key advantage of the algorithm is that no arbitrary assumption on the data distribution is required, as it relies purely on the proximity of the neighbor points. On the other hand, the algorithm is therefore sensitive to any incorrectly labeled entries in the training dataset that may lead to further spurious assignments.

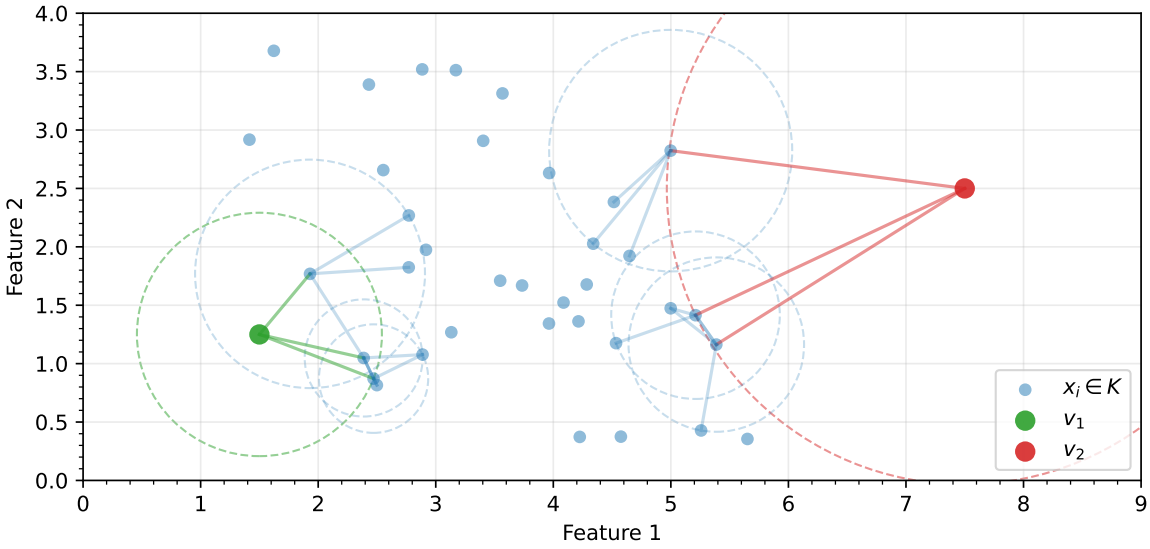


Figure 2.7: Idea of the Local Outlier Factor applied as an outlierness measure. The $k = 3$ closest neighbors are considered when identifying the reachability distances (represented as the radiuses of circles). For element v_1 the reachability distance is similar as for it's neighbors, whereas for element v_2 it is significantly greater.

It shall be noticed that for the very, very large datasets, containing huge number of high-dimensional feature vectors ($n \sim 1\,000\,000\,000$), the process of identifying the nearest neighbors can become slow and the model may be difficult to fit into RAM (Random Access Memory). To overcome this, companies that deal with searching through such enormous amounts of data on daily basis presented recently their dedicated solutions for this problem. The Faiss library[2] [19], developed mainly by Fundamental AI Research group at Meta (formerly Facebook), is an example solution that utilizes the product quantization based approach for approximate nearest neighbor search [39]. Another example is Annoy library[3] (Approximate Nearest Neighbors Oh Yeah), developed by Erik Bernhardsson and used at Spotify for music recommendations, that supports file-based indexes mapped into RAM that can be effectively shared between multiple system processes.

During the research the implementation from the scikit-learn library [51] was used.

2.3.5 Local Outlier Factor

The Local Outlier Factor (LOF), originally described by Breunig et al. [6], is a well-established algorithm widely used to detect abnormal data in high-dimensional spaces. It is based on the concepts of so called reachability distance and local reachability density. Instead of considering the global data distribution, it aims to identify outliers by analyzing only the local neighborhood. Hence, it can be considered as an extension of the k-Nearest Neighbors algorithm.

Let $N_k(v, T)$ be a function that returns the set containing k number of elements x_i from cluster T that are closest to v. The base concepts utilized by LOF are therefore defined as follows. First, the $kdist(v, T)$ is defined as the distance from given v to its k-th neighbor from the cluster T,

$$kdist(v, T) = \max\Big\{ \forall x_i \in N_k(v, T) : \ \Big\|\overrightarrow{v - x_i}\Big\| \Big\}. \tag{2.20}$$

Then, the reachability distance of element v with respect to the element x and cluster T is defined as either the true distance between v and x, or as the $kdist(x, T)$ distance of element x, whichever turns greater,

$$rd_k(v, x, T) = \max\Big\{ kdist(x, T), \ \Big\|\overrightarrow{v - x}\Big\| \Big\}. \tag{2.21}$$

Successively, the local reachability density for an element v with respect to the given cluster T is defined as the inverse of the average reachability distance of element v and its k neighbors from set T,

$$lrd_k(v, T) = \frac{k}{\sum_{x_i \in N_k(v,T)} rd_k(v, x_i, T)} \ . \tag{2.22}$$

Finally, the outlierness score for a given vector v against the target data cluster T is calculated as an average local reachability density of k neighbors of v, divided by the local reachability density of the v element,

$$LOF(v, T) = \frac{\sum_{x_i \in N_k(v,T)} lrd_k(x_i, T)}{k \cdot lrd_k(v, T)} \ . \tag{2.23}$$

[2] `https://github.com/facebookresearch/faiss`

[3] `https://github.com/spotify/annoy`

Figure 2.7 illustrates the idea of Local Outlier Factor algorithm. Intuitively, the algorithm compares the radiuses of circles corresponding to the reachability distances of the analyzed points and their k closest neighbors. Shall these radiuses be similar to their neighbors ones (v_1, score $LOF \approx 1$), the point can be considered an inlier. On the other hand, when the radius at analyzed point is much greater (v_2, $LOF \gg 1$), then such point is likely an outlier. High score value means a given point v is relatively far from the cluster, as there is low density of points in the surrounding area (inverse of reachability distance).

It is worth to mention that, similarly to k-Nearest Neighbors algorithm, the Local Outlier Factor does not require any *a priori* assumption on the data distribution, since it identifies the outliers only by analyzing the local surroundings of data. Hence LOF algorithm represents so called non-parametric approach to anomaly detection.

The LOF algorithm captures the intuition that outliers are isolated points, while denser regions of data are related with areas typical for a given distribution. It can be adapted to handle different distance metrics as well. Nevertheless, the value of k affects the sensitivity of the algorithm to any local variations in density. However, the algorithm can be computationally expensive for large datasets due to the complexity of finding neighbors in data.

During the research the implementation from the scikit-learn library [51] was utilized.

2.3.6 Mahalanobis distance

The Mahalanobis distance is one of the state-of-the-art solutions for performing out-of-distribution detection nowadays – often proposed directly as a method [43][21][20], mentioned in benchmarks [65][86][52][83] or used as a baseline when compared with new techniques [45][12][62]. It was originally described in 1936 by an Indian statistician P. C. Mahalanobis in [47] as a generalized distance metric for a normally distributed multidimensional data, i.e., multivariate Gaussian distribution. It is claimed to be effectively applicable to high-dimensional data and promised to be more accurate than other measures, especially if there are correlated features in data.

It is an example of parametric method for outlier detection, that assumes in advance a specific characteristic of the data distribution, here: MVN (Multivariate Normal distribution); and the parameters of such a distribution, such as vector of means (μ) and the covariance matrix (Σ), are estimated based on a provided training dataset.

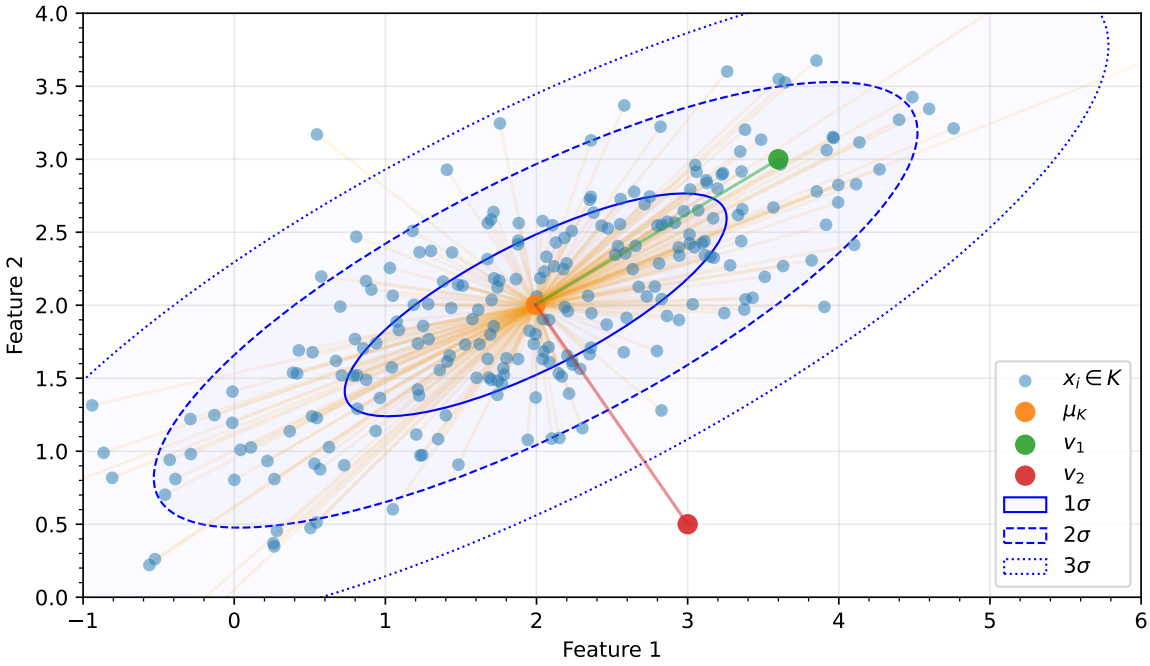


Figure 2.8: Idea of the Mahalanobis distance applied as an outlierness measure. The confidence ellipses indicate the distribution properties of cluster T; the marked areas correspond to regions in which about 68.2%, 95.4% and 99.7% data are located. Despite that elements v_1 and v_2 have similar Euclidean distances from the cluster center μ_K, only the element v_1 is located in the more typical region, still surrounded by elements $x_i \in T$, while v_2 shall be considered an outlier in this case.

The outlierness score for a given vector v against the data cluster T is calculated as

$$MD(v,T) = \sqrt{(\overrightarrow{v-\mu_T})^\top \cdot \Sigma_T^{-1} \cdot (\overrightarrow{v-\mu_T})}. \tag{2.24}$$

The μ_T represents the center of cluster T – it is a vector of means of each variable ($\mu_T \in \mathbb{R}^d$). The Σ_T^{-1} corresponds to the inverse of the covariance matrix Σ_T calculated for the cluster T. In statistics the Σ_T^{-1} element is also known as the precision matrix [80] (or concentration matrix).

High output values indicate the larger distance from the distribution center, suggesting that v is potential outlier. Contrary, low values imply that v resides close to the center of cluster T, conforming more to the data distribution and not being an outlier.

It shall be noticed that without the additional factor Σ_T^{-1} the formula 2.24 would be an equivalent to the classical Euclidean distance in $\mathbb{R}^d$ space. Hence, the Σ_T^{-1} element can be interpreted as a scaling factor of the space where the distance is measured. In other words, the distance between analyzed point v and the cluster center μ_T is adjusted

to account the distribution shape, as the straight Euclidean distance along one axis can be more typical than the same straight distance along the axis where the data are more concentrated.

Figure 2.8 illustrates such example, where the element v_1 can be considered as belonging to the data distribution, while the element v_2 is an outlier, despite the fact that both elements have similar Euclidean distance from the distribution center μ_T.

It must be taken into account that as the estimation of the covariance matrix is required, fitting the algorithm to the training data can be computationally expensive if the datasets is large (i.e., big number of samples, high dimension of feature vectors). It is worth to notice that the typically used algorithms, like Maximum-Likelihood Estimation (MLE), are sensitive to the presence of any outliers in the dataset, so in cases where any contamination in the dataset may be present the literature suggests utilizing other techniques, such as the Minimum Covariance Determinant estimator [58][51][4].

Additional requirement for the inverse of covariance matrix Σ^{-1} is that it must be positive-semidefinite and symmetric. The condition is that for a symmetric real matrix M of dimension $d \times d$ there exist no such vector $v \in \mathbb{R}^d$ that would produce a negative result of a product $v^\top \cdot M \cdot v$; formally

$$M = M^\top \ \wedge \ \forall v \in \mathbb{R}^d : v^\top \cdot M \cdot v \geq 0 \iff M \text{ is positive-semidefinite.} \tag{2.25}$$

When that condition is met, the M can be expressed as a product of a lower triangular matrix A and its transpose $A^\top$, i.e., $M = A^\top \cdot A$, known as the Cholesky decomposition [31]. Then, the product $v^\top \cdot M \cdot v$ is equal to the length of the vector v transformed by the matrix A,

$$v^\top \cdot M \cdot v = v^\top \cdot A^\top \cdot A \cdot v = (A \cdot v)^\top \cdot (A \cdot v) = \|Av\|. \tag{2.26}$$

Intuitively, this condition ensures that the output of formula 2.24 is not negative and can be interpreted as a distance metric. However, this condition cannot be satisfied when the number of samples n is lower than features space dimension d or the rank of the matrix Σ^{-1} is lower than d (as a result of highly correlated features or linearly dependent columns in the data – observed during research for ViT representation [section 2.4.7], further detailed study is needed).

[4] https://scikit-learn.org/stable/auto_examples/covariance/plot_mahalanobis_distances.html

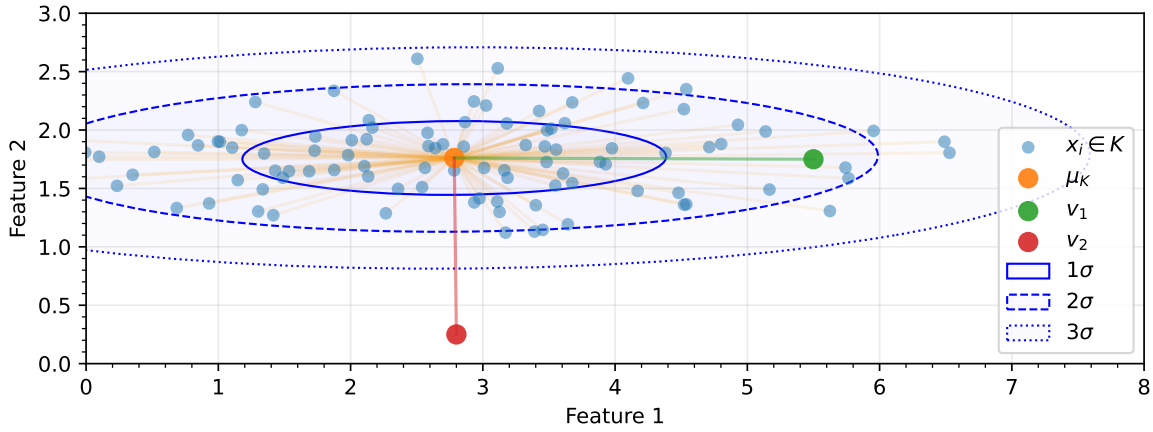


Figure 2.9: Idea of the Standardized Euclidean distance applied as an outlierness measure. Cluster T displays the heterogeneous spread along the axes – higher variance on feature 1 and lower variance on feature 2. Considering the different variances, after scaling the axes ($\sigma_1 \approx 1.5$, $\sigma_2 \approx 0.3$), the element v_1 is located closer to center μ_K than the element v_2 that is outlier (distance $d_1 \approx \frac{2.7}{1.5} \approx 1.8$, distance $d_2 \approx \frac{1.5}{0.3} \approx 5.0$).

Difficulties in practical applications for high-dimensional data involve the requirement of providing the sufficient number of training samples to fit the distribution model, i.e. at least $n_T \geq d$ and ideally $n_T \gg d$, for stable estimation of the covariance matrix Σ_T. However, for common datasets, such as *ImageNet-1k* [59], there are underrepresented categories present that not satisfy this requirement, especially for large models with $d > 1000$ number of features, e.g., ResNet ($d = 2048$ features), ConvNeXT ($d = 1536$ features), EfficientNet ($d = 1280$ features). Hence, the approaches emerged, such as proposed in literature [43][65], to calculate the single covariance matrix Σ using the whole training dataset for all classes as input, while still distinguishing means μ_T per data cluster during the distance calculation. This approach is sometimes called as utilizing the pooled covariance matrix [37][55] and so the outlierness score can be calculated here for m classes and vector of labels y as follows:

$$\Sigma = \frac{1}{n} \sum_{c=1}^{m} \sum_{i:y_i=c}^{n_c} (\overrightarrow{x_i - \mu_c}) \cdot (\overrightarrow{x_i - \mu_c})^\top, \tag{2.27}$$

$$MDP(v, T) = \sqrt{(\overrightarrow{v - \mu_T})^\top \cdot \Sigma^{-1} \cdot (\overrightarrow{v - \mu_T})}. \tag{2.28}$$

During the research the implementation from the SciPy library [70] was utilized.

2.3.7 Standardized Euclidean distance

The Standardized Euclidean distance is another measure based on Minkowski metric of order 2 in $\mathbb{R}^d$ space – a square root of the sum of vector elements to the power of 2. It considers axes-wise variances to normalize the contributions of each vectors elements when calculating the distance. It can be considered as a compromise between the traditional Euclidean distance, that assumes uniform relevance of all axes, and more general Mahalanobis distance, that estimates the distribution shape using the covariance matrix.

The outlierness score for a given vector v against the data cluster T is calculated as

$$SED(v,T) = \sqrt{\sum_{j=1}^{d} \frac{1}{V_{T,j}} \cdot (v_j - \mu_{T,j})^2}. \tag{2.29}$$

The V_K represents a vector of variances; the $V_{T,j}$ is a variance along j-th axis in cluster T and v_j is the j-th components of the vector v. The μ_T corresponds to the center of the cluster T. When the calculated value is high, the given data point is distant from the cluster center and is likely an outlier, while lower values indicate data close to the given cluster.

Similarly to the Mahalanobis distance, it scales the axes to have a uniform, unitary variance. However, contrary to the Mahalanobis distance, the Standardized Euclidean distance does not take into account the correlations between features in the data, assuming those are independent. Because of this, it is much more efficient when utilized in high-dimensional spaces, because no computationally expensive inversion of covariance matrix is necessary to fit the algorithm to the training data, while it still considers the orientation and shape of data distribution.

Furthermore, as discussed in the chapter 3 (section 3.5), the estimation of variances is much more stable than the estimation of covariances, so a lower method error would be made when the analyzed data are uncorrelated or the correlation of features is negligibly small.

During the research the implementation from the SciPy library [70] was utilized.

2.4 Learned representations of image and text data

Outlierness measures described in section 2.3 quantify the similarity of feature vectors v with respect to the target class c_K represented by the dataset cluster K. In essence, these are mathematical calculations performed on the vectors of real numbers. However, not every real data are useful vectors of real numbers applicable for such calculations.

For real-world applications, e.g., classification of text documents or images, these elements have to be first transformed from their original form to the feature vectors, utilizing so called vectorizers. There are multiple known algorithms for performing such transformation – some that produce simple to understand features as outcomes, e.g., counts of selected important keywords in text documents; and some that utilize sophisticated techniques, such as multi-layered convolutional networks, that produce feature vectors with not so transparent meanings for humans.

Chapter 4 is devoted to studying the performance of different OOD detectors in the feature spaces - representations of different Deep Learning models for image and text recognition. Here a brief summary of each algorithm is provided with references.

2.4.1 CLIP

The CLIP (Contrastive Language-Image Pre-training) offers a technique for processing the image into a feature vector that utilizes the a pre-trained convolutional neural network, which uses a transformer-based architecture.

Originally it is a deep learning model, developed by Alec Radford et al. [53] at OpenAI[5], that consists of two separate encoder networks, trained on pairs of images and their descriptions (text). Both images and texts are transformed into feature spaces that are then mapped into a common space, such that the similar images and their descriptions would be clustered together. In the result, the relationship between text and images is produced. The constructed model is capable of providing a description of the provided images, as well as identifying the images that matches the provided text.

2.4.2 CoCa

The CoCa (Contrastive Captioners) is another method that involves the transformer-based architecture, that can be utilized in generation of feature vectors from images.

[5] https://github.com/openai/CLIP

Described by Jiahui Yu et al. [84] at Google, it aims to further advance the capabilities previously seen in models such as CLIP – by co-training vision and language models together. It relies on a two–stage training process: first, a vision transformer and a text transformer are trained independently on pairs of images and captions; then, in the second, contrastive stage, the created transformers are fine-tuned together using a contrastive loss function that encourages similar representations for matching image-caption pairs. The produced model incorporates generative capabilities that allows to generate image captions with a natural language processing (NLP) techniques.

The CoCa model is available as a part of an open source implementation of CLIP[6].

2.4.3 ConvNeXT

The ConvNeXT is a pure convolutional model that can be used for obtaining feature vectors from images. It promises to be accurate, efficient and scalable while remaining very simple in design (contrary to transformers).

Proposed by Zhuang Liu et al. [46] at Facebook AI Research[7], it utilizes a modernized variant of a standard ResNet architecture, inspired by the design components of hierarchical vision transformers (Swin), without involving any attention-based modules that introduce quadratic complexity to the final model (with respect to the input size). During study the authors identified several key components that lead to increased performance on image recognition tasks, resulting in a family of models that are not only more efficient than standard convolutional approaches, but can even outperform the transformer-based architectures.

2.4.4 EfficientNet

The EfficientNet represents a family of the state-of-the-art classification models for images that utilize a convolutional neural network architecture and so called compound scaling method. It is also capable of providing feature vectors effectively.

Invented by Mingxing Tan and Quoc V. Le [66] at Google Research[8], if focuses on constructing a model that would achieve high classification accuracy, while remaining efficient to compute. They key lies in the utilization of a scaling method that uniformly escalate the network dimensions (depth, width and resolution) for provided training

[6] `https://github.com/mlfoundations/open_clip`

[7] `https://github.com/facebookresearch/ConvNeXt`

[8] `https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet`

data with fixed coefficients – balancing model accuracy and efficiency. This is justified by the intuition that for the larger input images the neural network also needs to be suitably bigger in order to maintain the accurate recognition ability. Additionally, it employs the squeeze-and-excitation technique to optimize the features representation. In effect, compared to historically previous approaches, it results in models that are smaller and faster to compute.

2.4.5 MobileNet

The MobileNet is a class of convolutional neural networks that prioritizes the reduction of computational cost at a trade off accuracy. Just like other neural networks, it is capable of producing feature vectors from images.

Developed by Andrew Howard et al. [35] at Google[9], it concentrates on building lightweight deep neural networks that would not be constrained by limited computational resources. While not aiming top-class accuracy, it still fall comparable to many popular models known in the literature. The reduced computational cost is achieved by replacing standard convolutional layers in the network architecture with depth-wise separable convolutions that significantly reduce the number of multiplications. This makes it well suited for applications on low-power computers, such as mobile phones and edge devices.

2.4.6 ResNet

The ResNet (Residual Networks) is yet another example of deep convolutional neural networks (CNN) architecture that can be utilized for automated pattern recognition and features extraction from images.

Proposed by Kaiming He et al. [25] at Microsoft Research Asia [10], it addresses the problem of vanishing gradient during backpropagation by introducing the concept of so called residual learning that involves shortcut connections, skipping one or more network layers. Such approach allows to construct much deeper models, i.e., containing more layers than were available with previously existing solutions, which effectively leads to more accurate representation of data and enables higher classification performance.

[9] https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet
[10] https://github.com/KaimingHe/deep-residual-networks

2.4.7 ViT

The ViT (Vision Transformer) represents a group of methods that use a transformer-based architecture for image classification. Although the approach is significantly different from the one used with convolutional neural networks, it still can be utilized in the process of features extraction from images.

Designed by Alexey Dosovitskiy et al. [18] at Google Research[11], it processes the input image into a number of patches that are converted into vectors and passed to a standard transformer encoder. Transformer is able to learn long-range dependencies and relationships between different parts of the image due to so called self-attention function. Before ViTs, such architecture was previously observed and successful in Natural Language Processing (NLP) applications. However, is turns out to be also well-suited for various computer vision tasks.

2.4.8 BERT

The BERT (Bidirectional Encoder Representations from Transformers) is a transformer model designed for NLP tasks. The same pre–trained model can be fine-tuned and used for a variety of specialized applications without requiring any substantial modifications to the architecture.

Published by Jacob Devlin et al. [17] at Google[12], it utilizes a multi-layer transformer encoder architecture. During the pre-training, it involves the Masked Language Modeling (MLM) and the Next Sentence Prediction (NSP). The MLM randomly hides words in the input sequence and attempts the model to predict it back, based on the remaining context. The NSP ensures the sentences are coming in proper consecutive order. This way the model learns the contextual representations of words and phrases from unlabeled text, without a need for any *a priori* language knowledge – just by analyzing the other words that come before and after in the sentences.

[11] `https://github.com/google-research/vision_transformer`
[12] `https://github.com/google-research/bert`

2.4.9 Doc2Vec

The Doc2Vec (Document to Vector) is an unsupervised algorithm to generate representations (feature vectors) from a collection of text documents, leveraging the word embeddings technique. It features automated identification of the relation between text terms and capturing of the semantic meaning, enabling tasks like document similarity retrieval and classification.

Developed by Quoc V. Le and Tomas Mikolov [42] at Google, it is built upon the concept of word embeddings learned by Word2Vec [49]. While Word2Vec focuses on individual words, the Doc2Vec addresses the whole text documents, aiming to represent documents as vectors in a high-dimensional space, where similar documents, regardless of their length, have similar vector representations.

2.4.10 TF-IDF

The TF-IDF (Term-Frequency/Inverse-Document-Frequency [41])[13] is a method of producing feature vectors from a collection of text documents – so called text corpus. It is similar to the Bag-of-Words technique, where the words occurrences are counted.

Just like in the BoW, the features in produced model correspond to the presence of selected words/terms. However, instead of just counting the occurrences, the TF-IDF involves a statistical weighting scheme that measures the significance of each word in a given corpus. By taking into account both: (1) how often the term appears in a single document, as well as (2) in how many other documents it occurs; it is capable of identifying features that would provide valuable contribution – reducing terms that appear too frequently in whole corpus, as they would not allow to distinguish specific elements or groups. Terms that are frequent within only a specific document but rare overall in the collection will have higher TF-IDF scores, indicating their relevance as the features.

It it worth to notice that, contrary to deep learning techniques, the TF-IDF benefits from the initial pre-processing of the corpora, such as filtering out the stop words and utilizing other Text Mining technologies to transform the words into their base forms – tokenization, stemming, lemmatization and POS (Part-Of-Speech) tagging [34][69].

[13] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer`

2.4.11 fastText

The fastText is a library that allows learning and producing the word embeddings (vector representations of words) from a provided text corpora and performing the classification of text documents. It utilizes a non-convolutional, shallow neural network.

Published by group of researchers, Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov [5][38] employed at Facebook AI Research[14], it further extends the techniques of Word2Vec [49]. Instead of analyzing just whole words, it rather considers the so called character n-grams – partial subwords; each word is represented by a sum of some character n-grams. This allows the model to also handle the out-of-vocabulary (OOV) data – previously unseen words, not encountered during the training, can still be represented as a combination of known character n-grams. Therefore the resulting model is versatile, while it also often remains lightweight enough for applications even on mobile devices.

[14] `https://github.com/facebookresearch/fastText`

Chapter 3

Performance of OOD detectors on the simulated data

This chapter contains the description of the conducted numerical study. The research was focused on comparing the performance of various algorithms (outlierness measures) for detecting the out-of-distribution data and their ability to properly recognize the data coming from the same distribution as the training set. The different properties of these algorithms were observed as an effect of considered parameters changes, such as the number of training samples, dimension of feature vectors, number of correlated features and the variance value of features. The performance was analyzed in terms of ID-OOD separability, expressed as AUROC score, as well as the classification results (sensitivity, specificity) using threshold calibrated at $95\%TPR$ with respect to the training data. Finally, performed the analysis of errors related to (in)accurate representations of training data by the outlier detectors, considering the influence of the feature space dimensionality and the number of samples. The involved data organization (number of training samples, dimension of feature vectors, Multivariate Normal Distribution) reflects the characteristics of (some) data from the subsequently analyzed real-world benchmark (chapter 4).

3.1 Baseline – samples, dimensions and distances

The goal of the first conducted experiment is to observe the behavior of selected outlierness measure and analyze their performance in the outlier detection task – and to establish the baseline for more specific examinations, performed in further sections. The effect of three major parameters is considered: number of training samples n, dimension of feature vectors d and the distance to out-of-distribution samples h; additionally utilized three different generator distribution functions to produce the data clusters – for even more versatile insight.

It should be noted that the organization of the simulated numerical study includes the ranges of n and d parameters values which are encountered in common OOD detection benchmarks for image and text recognition using Deep Learning models (chapter 4).

3.1.1 Experiment organization

The experiment is organized as follows:

- First, 3 data clusters are generated.
  - The set of training data T, representing the in-distribution (ID) data, containing n samples of dimension d, produced from a chosen generator G (*Gaussian*/*MVN*, *triangular* or *uniform* distribution – that is located around the center of the coordinate system $\mu = [0, 0, \ldots, 0]$ with spread of ± 1).
  - The set of known data K, representing a testing dataset (another examples of ID data), generated from the same distribution as T, with a fixed number of 1000 samples. It is used to analyze the sensitivity of the detector, i.e., the ability to properly recognize testing data as similar to the training data.
  - The set of unknown data U, representing out-of-distribution (OOD) data and consisting of a fixed number of 1000 samples, produced by the same generator as T, however with the distribution center shifted by the distance h in space (so the mean is at location $[\frac{h}{\sqrt{d}}, \frac{h}{\sqrt{d}}, \ldots, \frac{h}{\sqrt{d}}]$). It is used to evaluate the specificity of the algorithm (i.e., proper detection of OOD samples).
- The selected algorithm OF (Outlier Factor) is fitted to the training dataset T.
- Next, the outlierness scores are calculated for each element of sets T, K and U.
- The separability between clusters K and U, using the selected OF, is analyzed by calculating the Area Under the Receiver Operating Characteristic (AUROC).

- The classification of data from clusters K and U with respect to the dataset T and outlierness measure OF is performed, using the threshold value t selected as the 95th and the 99th percentile of outlierness scores obtained for the cluster T.

Summarizing, the input parameters that vary in the experiment are: number of training samples n, dimension of feature space d, distance to the outliers h, outlierness measure OF and the generator distribution G.

Additionally, for each combination of parameters, the experiment was repeated several times with various values of the generator seed ξ (that affected the values within T, K and U) to observe the variability of results.

3.1.2 Experiment results – distribution properties

The aim of the first experiment was to analyze the behavior and usability of selected outlierness measures, discussed in section 2.3, especially when applied in high-dimensional feature spaces, considering such factors as the number of training samples used to model the in-distribution data.

The study shows that various techniques used to model the in-distribution data, such as Euclidean distance (ED, section 2.3.2), Integrated Rank Weighted Depth (IRWD, section 2.3.3), k-Nearest Neighbors (kNN, section 2.3.4) and Local Outlier Factor (LOF, section 2.3.5), have completely different properties. Although all of them can be considered as distance metrics, their output values are not directly comparable, e.g., the same element v can obtain score $s = 20$ using Mahalanobis distance (MD, section 2.3.6) and score $s = -0.65$ with Angle-Based Outlier Factor (ABOF, section 2.3.1). Hence, selecting any arbitrary threshold value t for distinguishing outliers, without additional analysis, is not universally possible.

Figure 3.1 presents example distributions of scores obtained for six different outlierness measures: ABOF, ED, IRWD, kNN, MD and LOF. First major difference between the techniques that can be noticed is in the ranges of values – as ED, kNN and MD are directly related to the spatial distances, the values are greater than for ABOF, IRWD and LOF, that rely on other quantities (variances scaled by distances, depth estimations with projections and comparison of neighbors' local reachability densities). The results obtained for Standardized Euclidean distance (SED, section 2.3.7) are nearly the same as for ED, due to the variance set to 1.0 in the experiment, hence they are omitted and not presented in figure 3.1.

It is worth to notice that from all considered measures only the IRWD is characterized by a limited range of the function value domain. For all other measures, there is either no upper limit or no lower limit.

- ABOF: $s \in (-\infty, 0.0]$,
- IRWD: $s \in [-0.5, 0.0]$,
- ED, kNN, LOF, MD, SED: $s \in [0.0, +\infty)$.

Note that the score values of ABOF and IRWD are negative in this study, because in implementation (Appendix B) the returned values of formulas 2.9 and 2.17 are inverted (multiplied by -1) to satisfy the criteria given by formula 2.3 (section 2.2.2) – i.e. having greater values to indicate the outliers.

For most of measures (ED, IRWD, kNN, MD) the distributions appear symmetrical. However, in case of LOF the positive skew can be observed (the longer tail is on the right) in case of in-distribution data (train and known data in figure 3.1e). Similarly, for ABOF the distributions are characterized by the negative skew (longer tail on the left side), especially in case of the distant out-of-distribution examples (unknown data in figure 3.1a). This phenomenon is more clearly visible in figure 3.2.

The most important observation here is that for some algorithms, notably kNN and MD, the observed outlierness score values for known in-distribution samples (cluster K, i.e. testing data) appear surprisingly distant from the values obtained for the training samples (cluster T). This means, that considering only the training samples' perspective (green barplots in figures 3.1d and 3.1f), most testing data coming from exactly the same distribution (blue barplots in the same figures) would be considered as out-of-distribution, i.e., outliers.

This phenomenon appears algorithm-specific and is observed regardless of generator distribution G, i.e., for *Gaussiann* (figures 3.1d and 3.1f), *Triangular* (figures 3.3c and 3.3d) and *Uniform* (figures 3.3d and 3.3f). For both kNN and MD the effect intensifies in high-dimensional feature spaces (figure 3.6). For MD the effect can be suppressed by increasing the number of training samples n, as visible in figure 3.4, however for kNN the increased n does not impact this phenomenon significantly (figures 3.5a, 3.5b and 3.5c). Inn case of kNN, the effect is reduced when greater number of k neighbors is considered (figure 3.5d).

The same effect can be observed for ED, SED and IRWD measures when the number of training samples n is lower than the dimension of features space d, as visible in the figure 3.9 – just like for MD, increasing the number of training samples n causes scores for in-distribution data (clusters K and T) to overlap. It is caused by difficulty

of obtaining accurate representation of a cluster in high-dimensional features spaces, discussed further in section 3.4. However, surprisingly, this effect was not observed in case of ABOF and LOF measures, even in case of extreme conditions, such as dimension of feature vectors $d = 5000$ and number of training samples $n = 50$, like shown in the figures 3.8.

Despite that the training samples (cluster T) may appear distant from the known in-distribution samples (cluster K), in all discussed cases there is a possibility to achieve a good separation between in-distribution data and outliers (cluster U). Hence, the Receiver Operating Characteristic (ROC) curves presented in figure 3.10 look similar – they present the relation between the sensitivity (True Positive Rate – TPR) and risk of type I error (False Positive Rate).

The ideal separation is reached in case of correct recognition of all in-distribution data without any spurious assignments of outliers – corresponding with top-left corner in ROC plot ($TPR = 1$, $FPR = 0$). The optimal threshold point marked in ROC plot is related to the threshold value that is closest to ideal situation (top-left ROC corner) – represented with the red cut-off vertical lines in figure 3.1. The second marked threshold, TPR95, corresponds to a cut-off value for which the 95% of in-distribution data were properly recognized.

The commonly used measure to evaluate the performance of classification model is the calculated Area Under the Receiver Operating Characteristic (AUROC) curve, with ideal value being $AUROC = 1.0$; any value $AUROC \leq 0.5$ means the classifier is worse than randomly performed assignments. In figure 3.10 all AUROCs are greater than 0.9, indicating very well separation between clusters K and U.

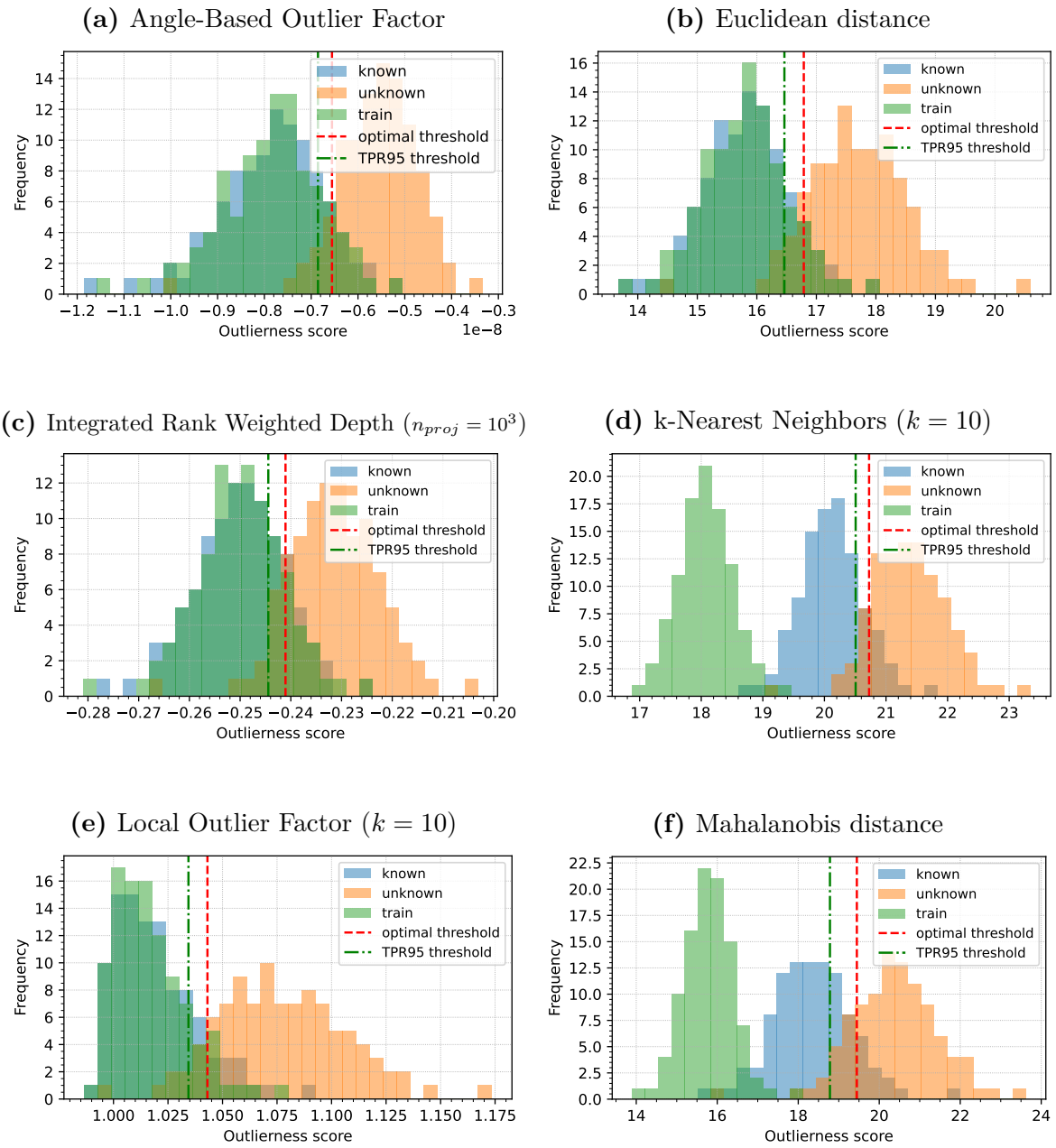


Figure 3.1: The distributions of outlierness scores obtained for various OF measures (ABOF, ED, IRWD, kNN, LOF, MD). For all cases the same configuration of T, K and U clusters is used – containing $n = 1000$ training samples, dimension of feature vectors $d = 250$, generated from $G = Gaussian$ distribution, seed $\xi = 0$; outliers are shifted by distance $h = 8$. In some cases (kNN, MD) the results obtained for K are surprisingly distant from results obtained for T.

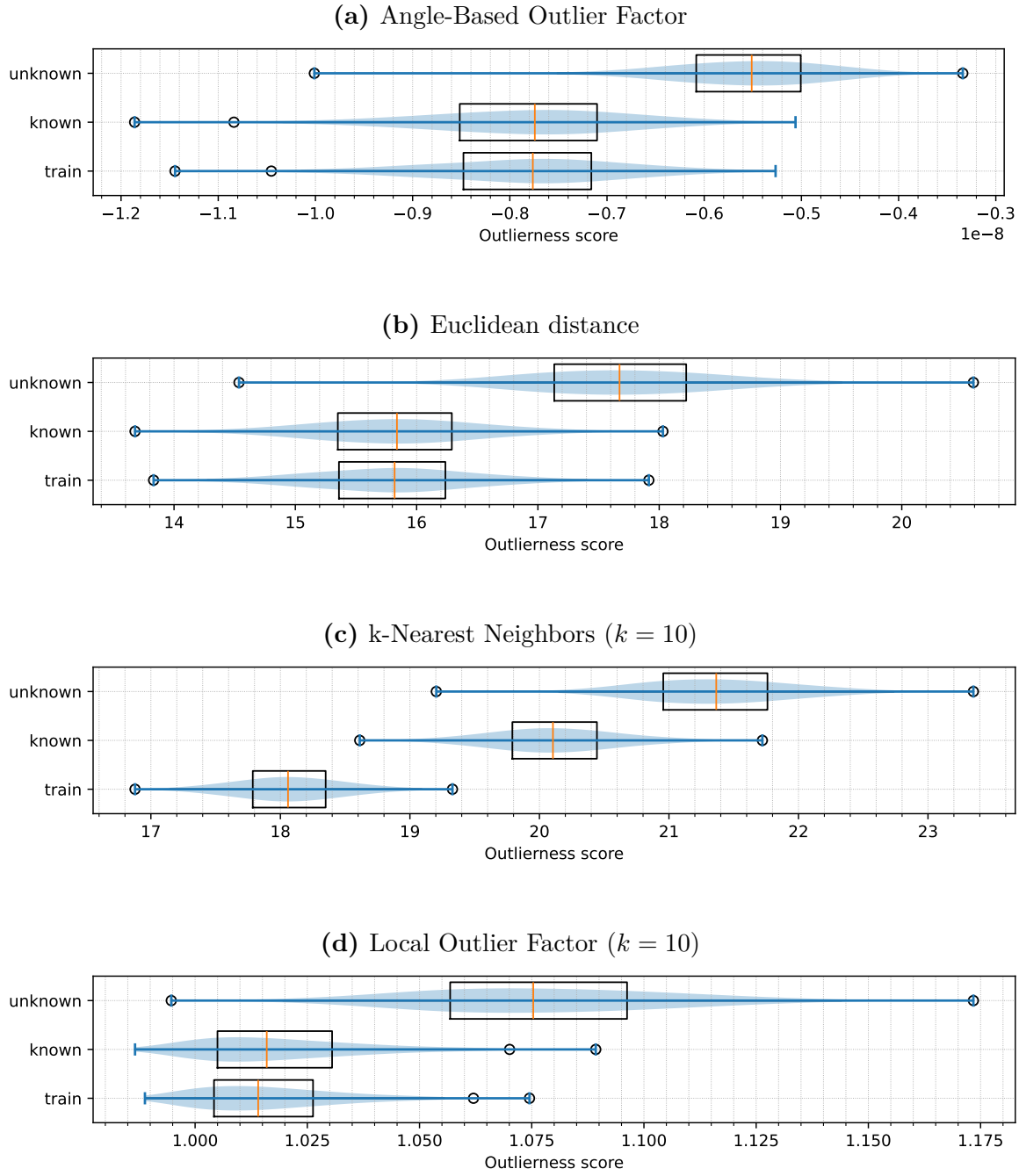


Figure 3.2: The boxplots of scores distributions obtained for selected OF measures (ABOF, ED, kNN, LOF) calculated on T, K and U clusters – corresponding with selected histograms from the figure 3.1. The positive skew is observed in case of LOF and negative skew in case of ABOF measure, while ED and kNN appear symmetric.

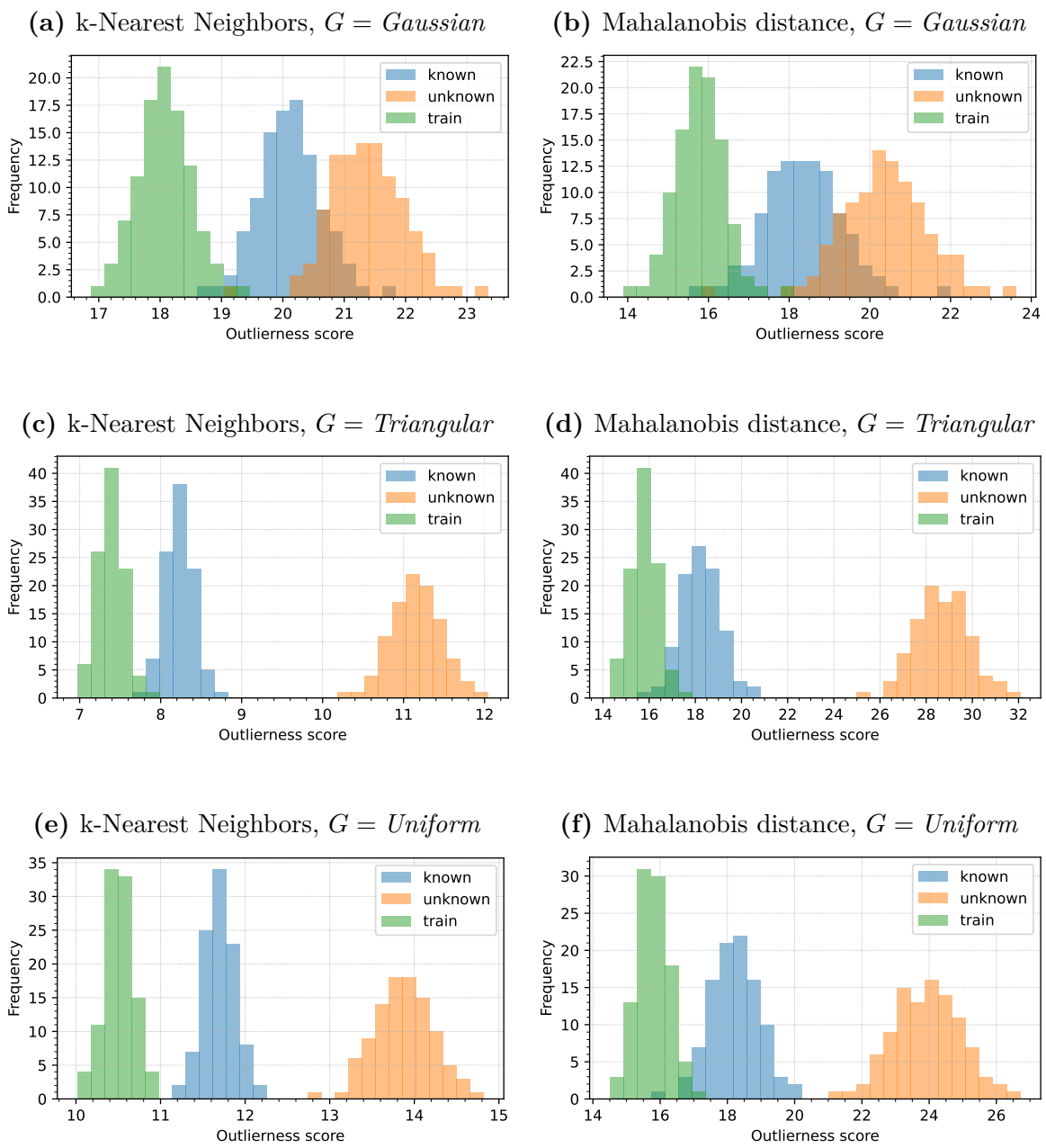
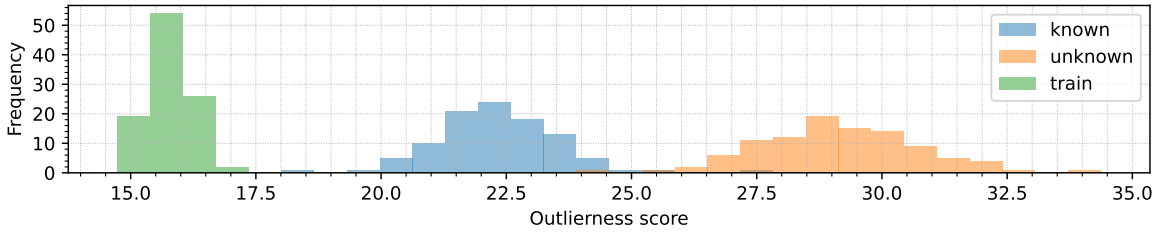
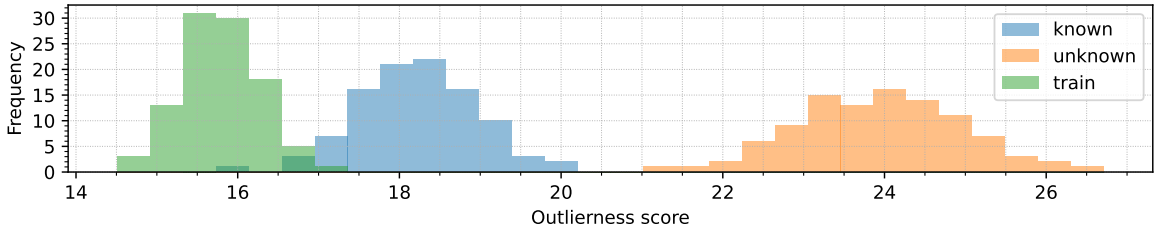


Figure 3.3: In case of kNN and MD, for high-dimensional feature vectors, the scores for known in-distribution data (cluster K) may not overlap with the scores obtained for the training samples (cluster T). This phenomenon is observed regardless of chosen data distribution generator G: *Triangular* (shown in figures 3.3c and 3.3d) or *Uniform* (figures 3.3e and 3.3f) *Gaussian* (figures 3.1d and 3.1f). Other parameters are the same as for figure 3.1: $n = 1000$, $d = 250$, $h = 8$, $\xi = 0$.

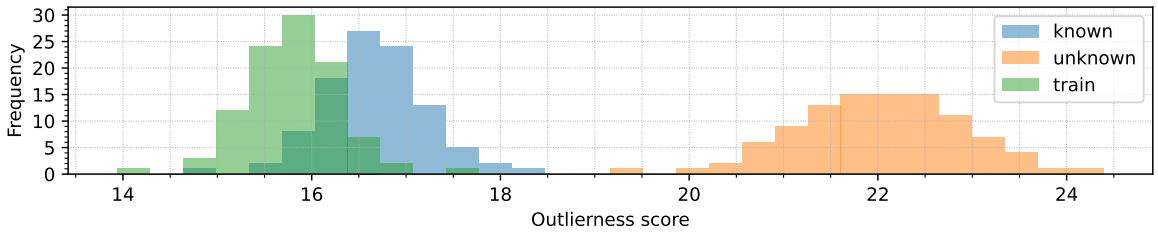
(a) Mahalanobis distance, training samples: $n = 500$



(b) Mahalanobis distance, training samples: $n = 1000$



(c) Mahalanobis distance, training samples: $n = 2500$



(d) Mahalanobis distance, training samples: $n = 5000$

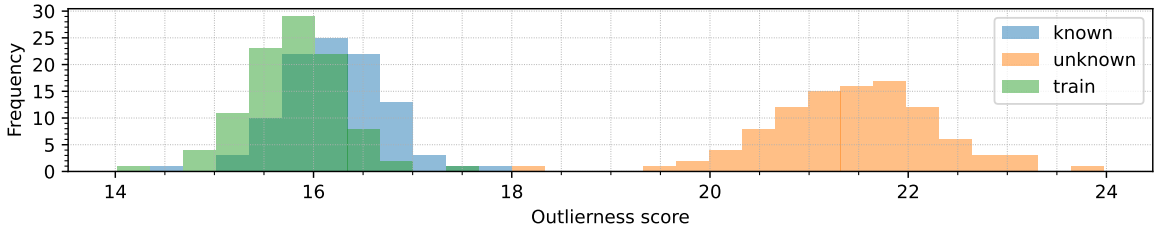
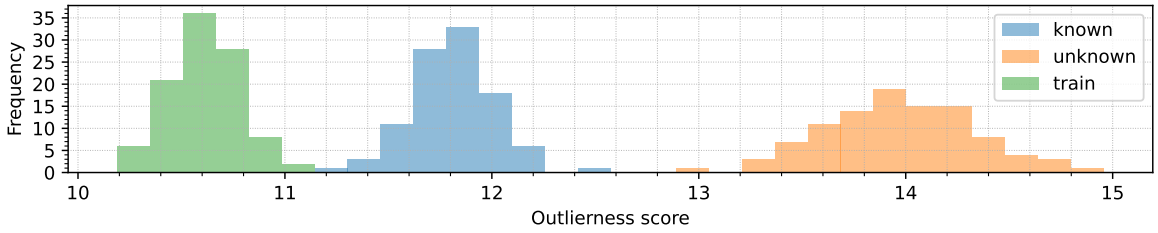
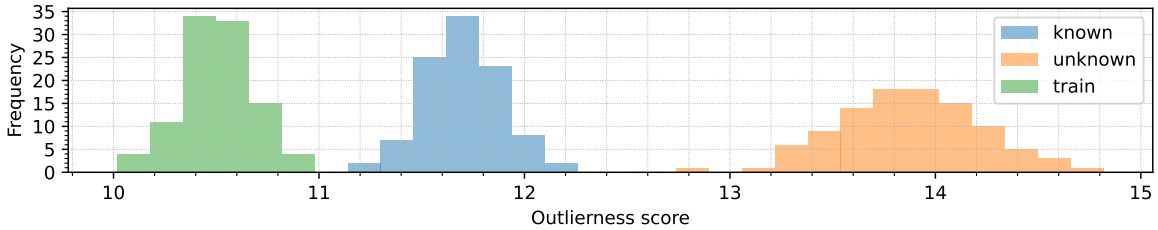


Figure 3.4: The distance between scores for known data (in-distribution, cluster K) and training examples (cluster T) gets smaller for MD when the training cluster T contains more elements (parameter n). Note that the outliers (unknown examples, cluster U) are also moving closer to T (distances between medians $Q2_T$ and $Q2_U$: $\Delta Q_{n=500} \approx 13.31 \rightarrow \Delta Q_{n=1000} \approx 8.19 \rightarrow \Delta Q_{n=2500} \approx 6.22 \rightarrow \Delta Q_{n=5000} \approx 5.68$), up to a certain point – when K overlaps with T, then cluster U no longer moves towards cluster T. Other distribution parameters involved: $d = 250$, $h = 8$, $G = Uniform$, $\xi = 0$.

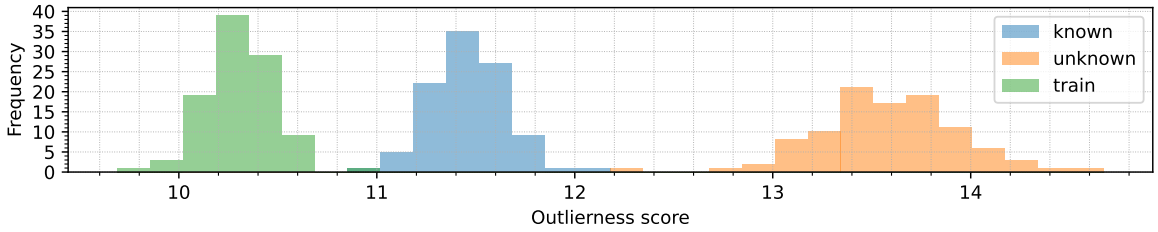
(a) k-Nearest Neighbors ($k = 10$), training samples: $n = 500$



(b) k-Nearest Neighbors ($k = 10$), training samples: $n = 1000$



(c) k-Nearest Neighbors ($k = 10$), training samples: $n = 5000$



(d) k-Nearest Neighbors ($k = 20$), training samples: $n = 1000$

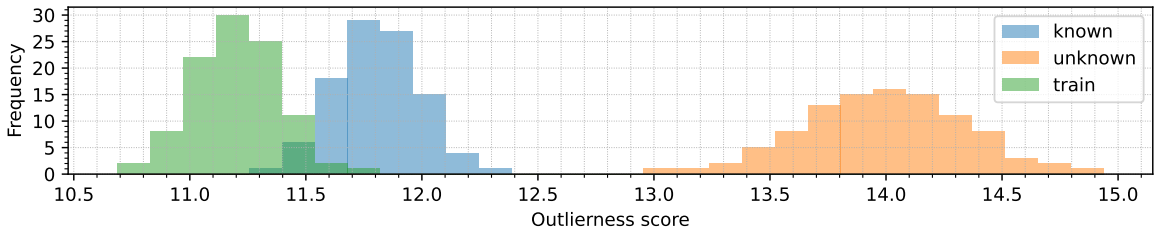
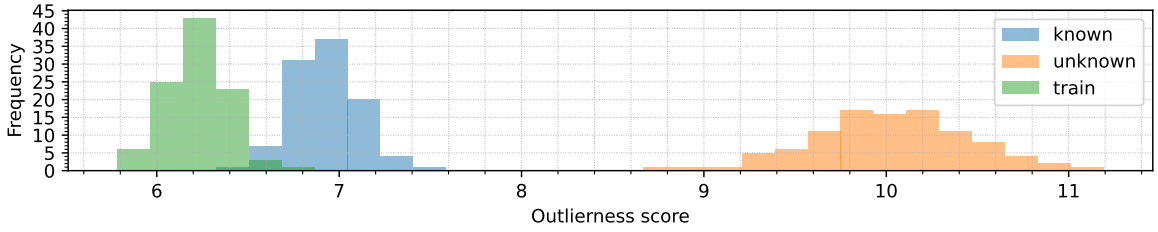
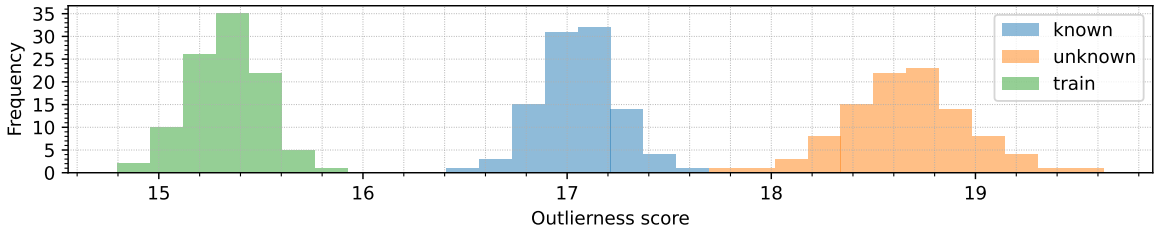


Figure 3.5: Unlike for MD, increasing the number of training samples n in cluster T does not bring the cluster K scores significantly closer to scores for cluster T. Scores obtained for outliers (cluster U) also remain unaffected by n. However, the results for K and T start to overlap for larger values of k (parameter of kNN algorithm). Experiment settings are the same as in figure 3.4 ($d = 250$, $h = 8$, $G = \mathit{Uniform}$, $\xi = 0$).

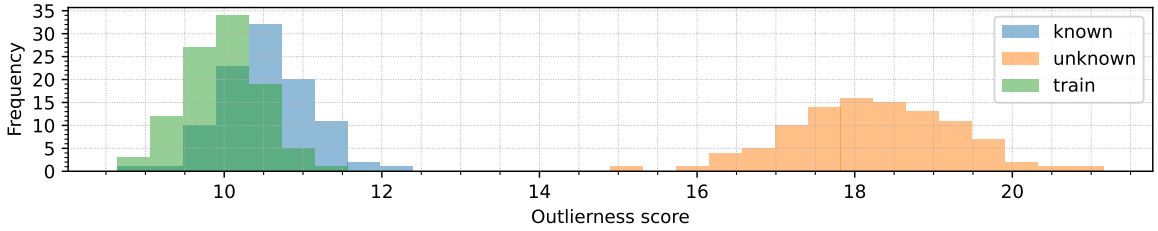
(a) k-Nearest Neighbors ($k = 10$), dimension of feature vectors: $d = 100$



(b) k-Nearest Neighbors ($k = 10$), dimension of feature vectors: $d = 500$



(c) Mahalanobis distance, dimension of feature vectors: $d = 100$



(d) Mahalanobis distance, dimension of feature vectors: $d = 500$

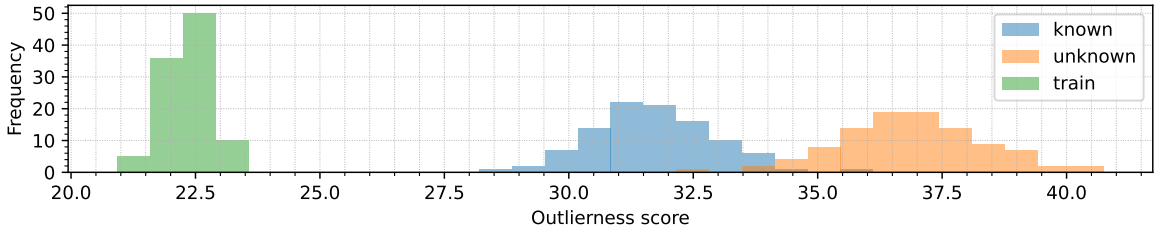
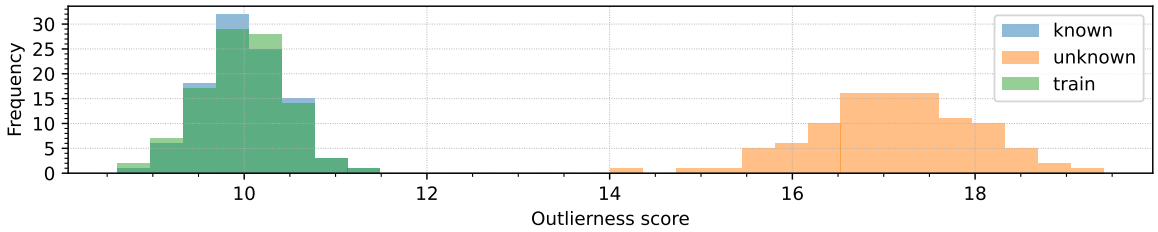


Figure 3.6: The effect of distancing scores acquired for cluster K from the scores obtained for cluster T, observed in case of kNN and MD measures, is stronger for increased dimensionality of feature vectors d. It can be noticed that for higher dimensions the scores values are also greater, as both the measures are based on spatial distances in features space, hence more feature vectors components contribute to greater score values. Results visible in plots are obtained for experiment settings: $n = 1000$, $h = 8$, $G = \mathit{Uniform}$, $\xi = 0$.

(a) Standardized Euclidean distance, dimension of feature vectors: $d = 100$



(b) Standardized Euclidean distance, dimension of feature vectors: $d = 500$

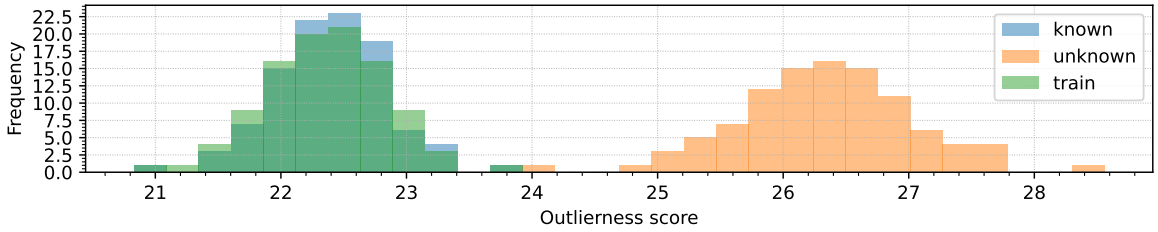
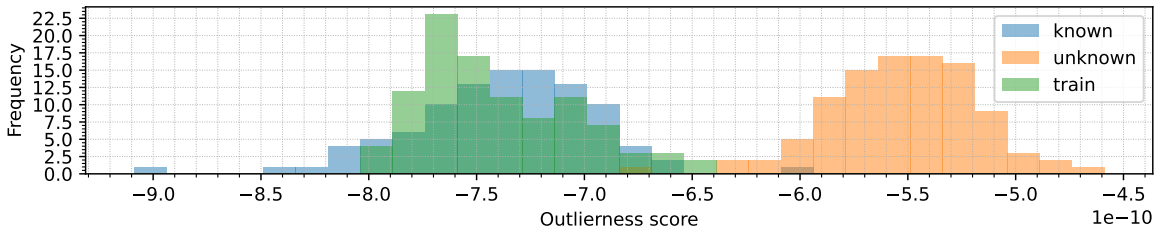


Figure 3.7: For measures ABOF, IRWD, LOF, ED and SED, in typical conditions, $n \gtrsim d$, the separation between scores for in-distribution data (cluster K and cluster T) is not observed, maintaining good overlapping even for a lower number of training samples n than for MD. Experiment settings: $n = 1000$, $h = 8$, $G = \mathit{Uniform}$, $\xi = 0$.

(a) Angle-Based Outlier Factor, dimension: $d = 1000$, samples: $n = 50$



(b) Local Outlier Factor ($k = 10$), dimension: $d = 1000$, samples: $n = 50$

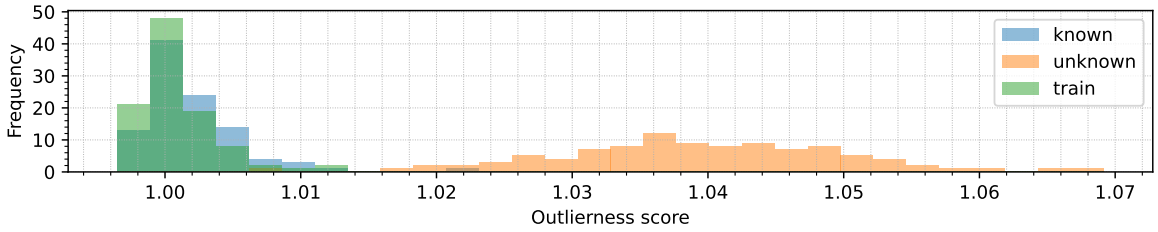
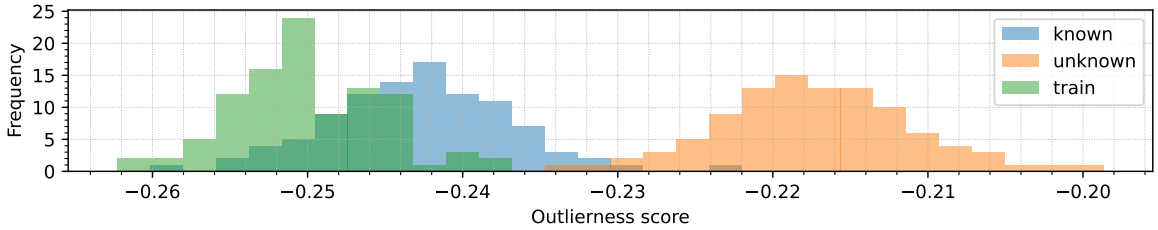
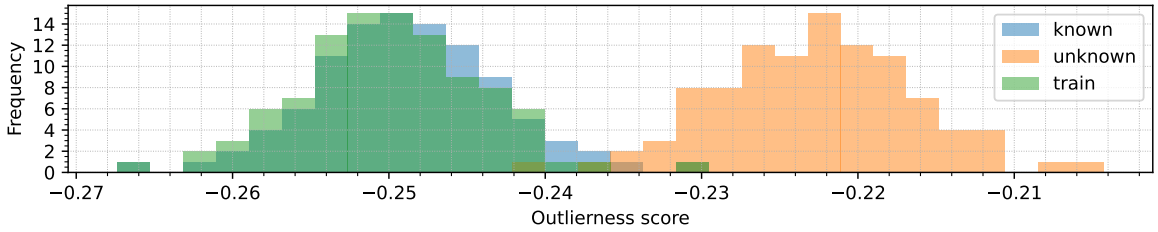


Figure 3.8: In the performed study, ABOF and LOF were able to produce accurate representations even in case of significantly under-represented testing cluster – obtained scores for T and K do overlap despite $n = 50$ training samples for $d = 1000$ dimension of feature vectors. Remaining experiment settings are: $h = 8$, $G = \mathit{Uniform}$, $\xi = 0$.

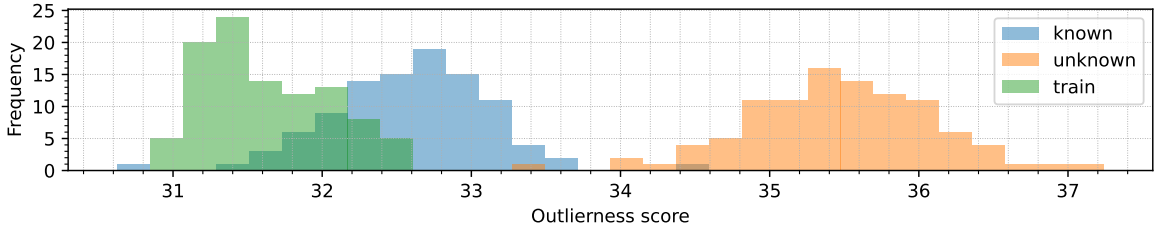
(a) Integrated Rank Weighted Depth ($n_{proj} = 10^3$), dimension: $d = 500$, samples: $n = 50$



(b) Integrated Rank Weighted Depth ($n_{proj} = 10^3$), dimension: $d = 500$, samples: $n = 500$



(c) Standardized Euclidean distance, dimension: $d = 1000$, training samples: $n = 50$



(d) Standardized Euclidean distance, dimension: $d = 1000$, training samples: $n = 1000$

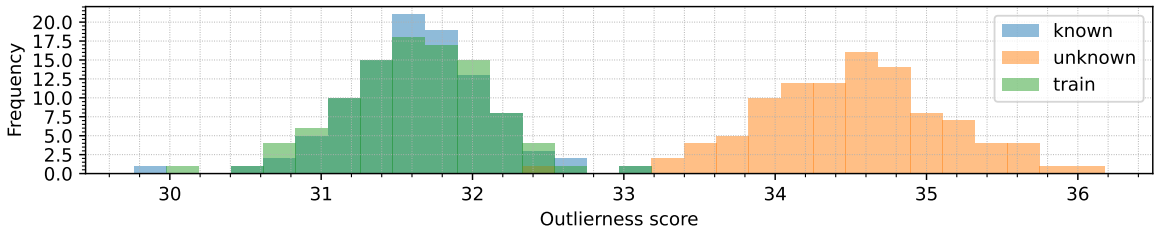


Figure 3.9: For strongly under-represented training clusters, $n \ll d$, the effect of not-overlapping between the scores for cluster T and K is observed in case of IRWD, ED and SED measures. The effect vanishes when n is not so low, yet it does not need to be as big as for MD to reach overlapping (settings: $h = 8$, $G = \mathit{Uniform}$, $\xi = 0$).

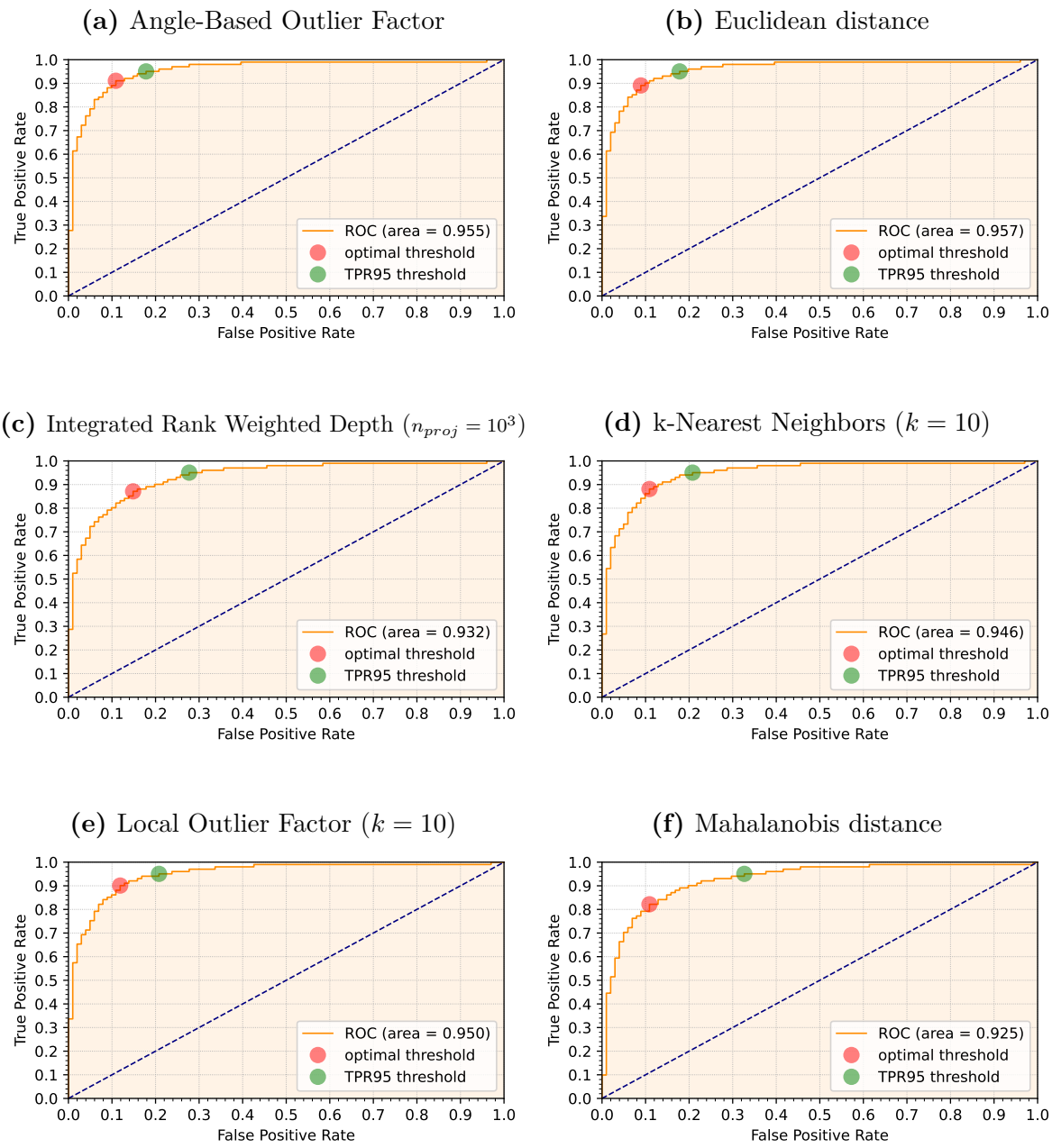


Figure 3.10: The Receiver Operating Characteristic (ROC) curves obtained for various OF measures (ABOF, ED, IRWD, kNN, LOF, MD). They show the sepearability between clusters K and U visible in corresponding plots from the figure 3.1. Despite that some OF methods represent K as distant from T, they still can distinguish between K and U quite well, all acquiring high AUROC scores (the area values of visible subplots are all greater than 0.9).

3.1.3 Experiment results – effects of parameters

Figure 3.11 illustrates how the performance of outlierness measures is affected by the dimension of the feature vectors d, under fixed number of training samples $n = 2500$ and distance to outliers $h = 8$. The experiments shows that higher the feature space dimension d, the more challenging the comparison between data vectors is, as both classification accuracy and AUROC score decrease. The research was focused on ED, IRWD, kNN, LOF, MD and SED measures, omitting ABOF as computationally too expensive and impractical for usage (primarily due to n; although it was reaching promising top-scores for lower n and d).

All of the analyzed measures OF provide good separability of in-distribution data and outliers in lower dimensions – reaching AUROC value above 0.95 for $d \leq 200$. In higher dimensions, the outliers distribution appear closer to the training data, so the obtained AUROC values are lower, decreasing exponentially with d. For dimension $d = 1000$ the AUROC reaches about $\sim$0.85 in case of ED and SED (plots overlap in figure 3.11a), $\sim$0.84 for kNN and LOF, $\sim$0.78 for MD and $\sim$0.715 in case of IRWD (results visible in subfigure 3.11a).

Although offering good separability, when the measures are involved in the classification task with respect to the training data only, not all OFs perform so well. Notably the kNN's performance falls drastically, reaching accuracy of $\sim$0.5 for $d \geq 250$. Similarly, the MD measure, after initially performing well ($d \leq 250$), shows gradual decay of accuracy in higher dimensions ($d \geq 500$). In both mentioned cases it is related to the lose of sensitivity, as visible in figure 3.11c – at some point all in-distribution data were recognized as outliers by kNN and MD (due to the same effect discussed in previous subsection 3.1.2 and visible in figure 3.6).

Contrary, in case of ED, SED, IRWD and LOF the lowered accuracy in high-dimensions is related purely to the decaying specificity – some out-of-distribution data are seen as too close to the training data, such as in histograms in previous subsection 3.1.2 (subfigures 3.1b, 3.1c and 3.1e), hence spuriously considered as inliers.

Figure 3.12 shows analogous research, analyzing the performance of outlierness measures OF affected by the number of training samples n, having fixed dimension of the feature vectors $d = 750$ and distance to outliers $h = 8$. Surprisingly, no strong influence between the accuracy and separability (AUROC value) is observed – except for extremely underrepresented cases ($n < 100$ for $d = 750$, not show in the figure) or Mahalanobis Distance measure.

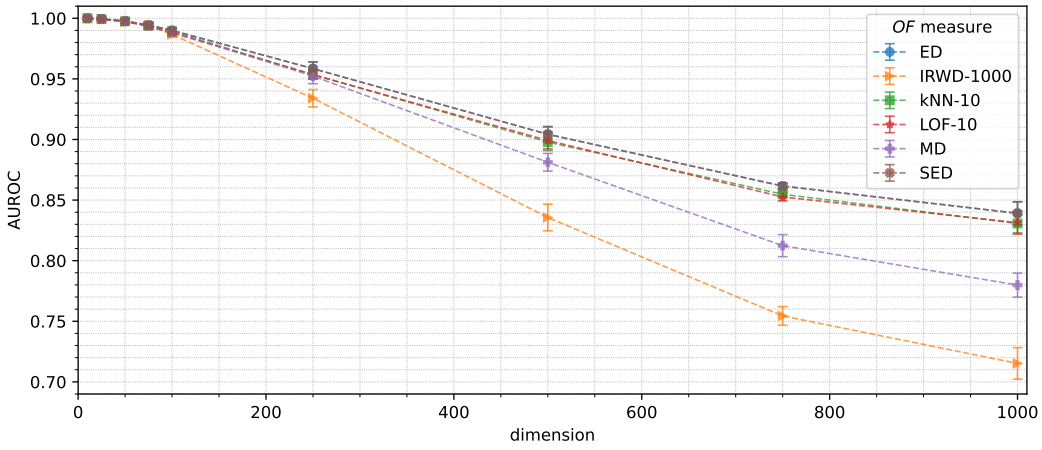
The estimation of covariance matrix for MD in features space of dimension $d = 750$ requires at least $n \geq 750$ data samples. Hence, first reasonable result for MD visible in figure 3.12a appears for $n = 1000$ – AUROC value ∼0.715; for $n = 750$ samples the AUROC is ∼0.5. For $n = 10000$ samples the reaches close to top AUROC score ∼0.85.

The best separability in analyzed case is again observed for ED and SED measures (AUROC values ∼0.86), then kNN and LOF (AUROC values ∼0.85). The IRWD performed significantly worse, scoring AUROC value ∼0.76. Similarly like in previously examined case, the good AUROC score does not translate to good accuracy in the classification task with respect to the training data. Again, it is due to the zero TPR score (sensitivity) – all in-distribution data are incorrectly recognized as outliers, because the outlierness scores for testing set do not overlap the scores for training set (as visible in figure 3.6). In case of MD, low accuracy is observed for up to $n \leq 2500$ samples, because of the same effect as for kNN, however by providing more training samples ($n \geq 5000$) the scores for in-distribution testing samples start to overlap with scores for training samples (effect visible in figure 3.4), in the end obtaining one of the top accuracy for $n = 10000$. Yet, ED, SED and LOF reach similar accuracy for lower number of training samples n.

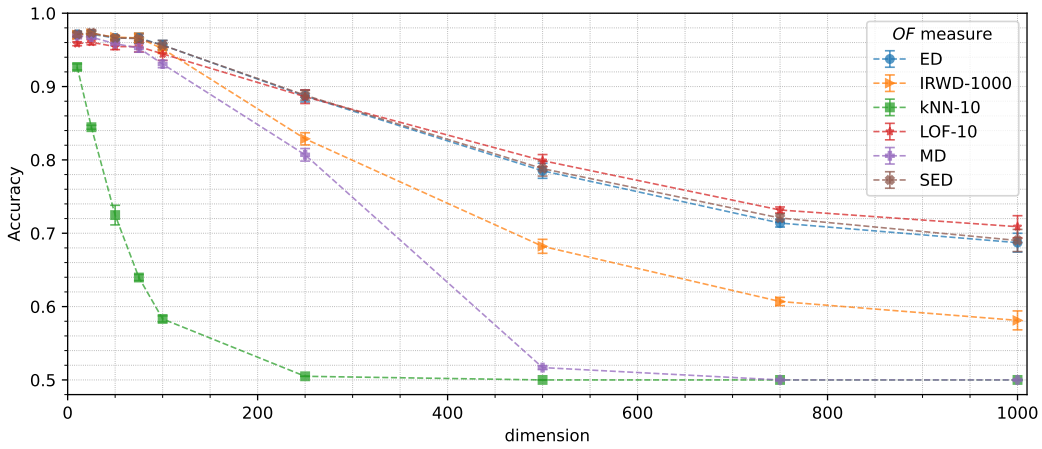
Finally, figure 3.13 presents the performance of outlierness measures OF as affected by the distance to outliers h, under fixed dimension of the feature vectors $d = 750$ and number of training samples $n = 2500$. Intuitively, the more distant the outliers are, the easier they are separable and detectable, like discussed in section 2.1.2 (Near OOD vs Far OOD). The relation is analogous as in first analyzed case – resulting in best separability for ED, SED, kNN and LOF, then for MD and worst for IRWD under given experiment configuration. Again, the worst possible accuracy is observed for kNN and MD, as for given n and d all the data are recognized as outliers (zero sensitivity). The ED, SED, IRWD and LOF initially do not recognize any outliers (zero specificity), until they are significantly distant $h > 4$.

The experiments were repeated for various distributions generators ($Gaussian$, $Triangular$, $Uniform$), however no significant differences were observed – both the classification and separability is easier (i.e., higher scores obtained) for a given set of parameters in case of the distributions with finite output domain ($G = Triangular$, $G = Uniform$), due to outliers appearing more distant, as seen in figure 3.3, however the overall trends and behaviors of measures remain similar. Hence, only the results for $G = Gaussian$ distribution are presented in this section. Additionally it will make it easier comparable with following results in sections 3.2 and 3.3. All omitted results can be analyzed in the tooling described in appendix B.

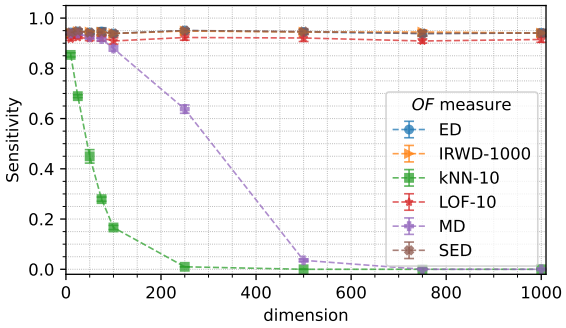
(a) Separability between in-distribution and out-of-distribution data



(b) Classification with respect to the training data



(c) Correctly recognized in-distribution



(d) Correctly recognized out-of-distribution

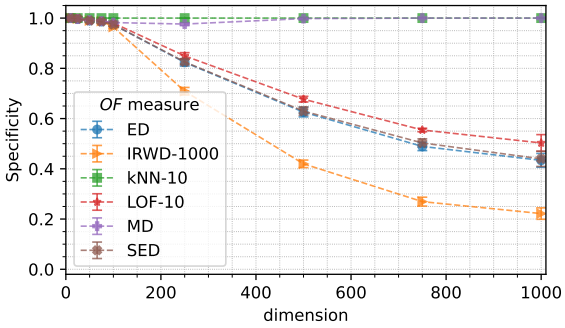
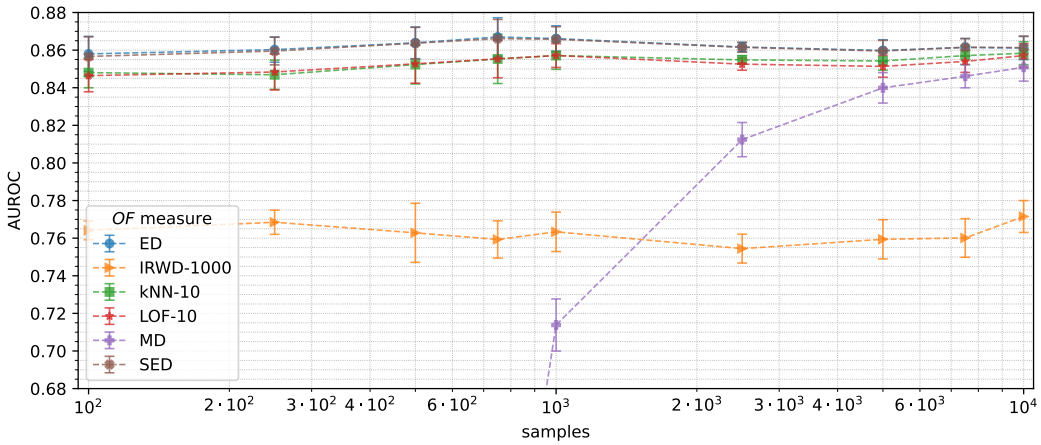
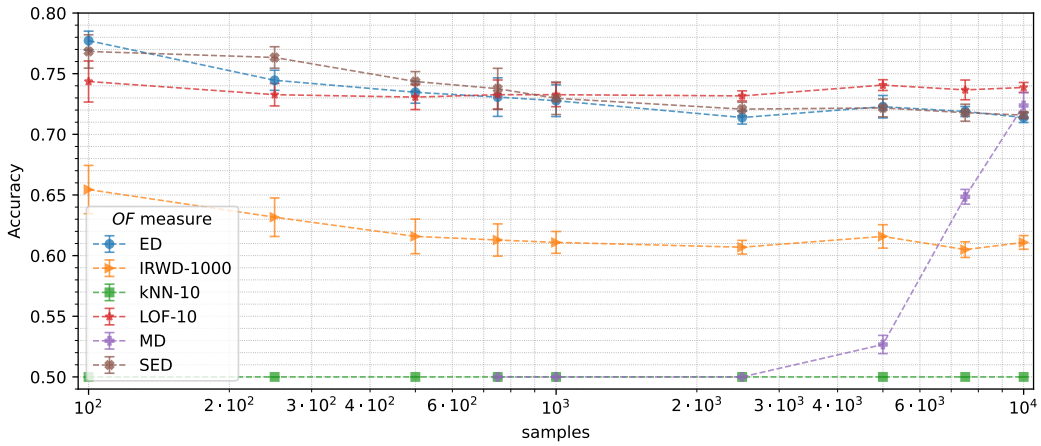


Figure 3.11: The performance of outlierness measures OF as affected by the dimension of the feature space d. The fixed parameters in the experiment are: number of training samples $n = 2500$, distance to outliers $h = 8$ and distribution $G = Gaussian$. The results are aggregated for multiple generator seeds ξ and displayed as averages with error bars (standard deviation).

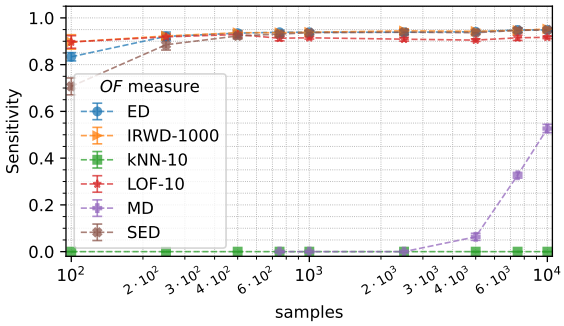
(a) Separability between in-distribution and out-of-distribution data



(b) Classification with respect to the training data



(c) Correctly recognized in-distribution



(d) Correctly recognized out-of-distribution

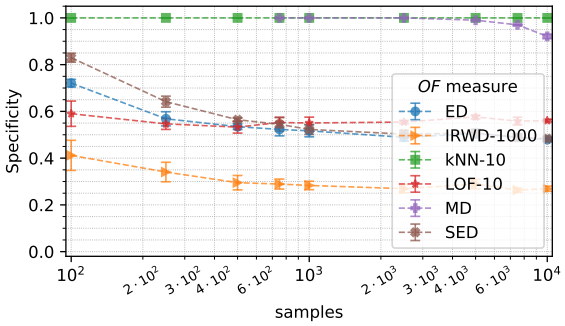
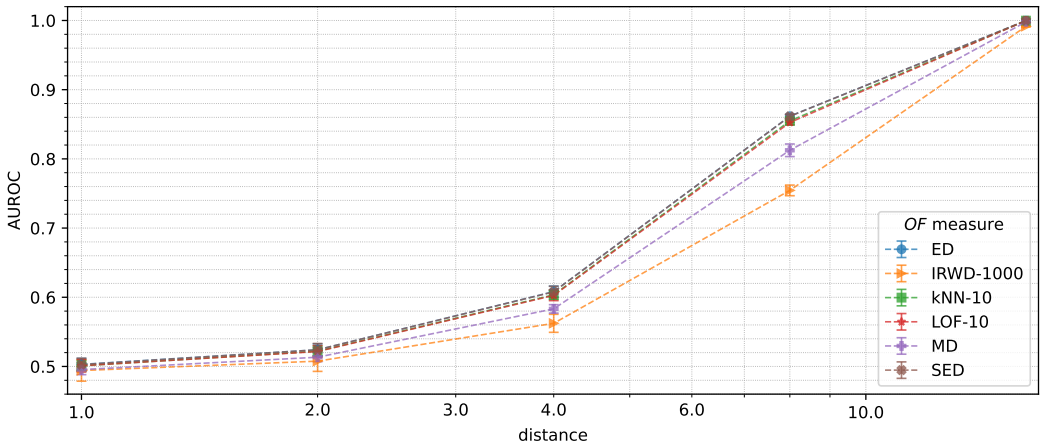
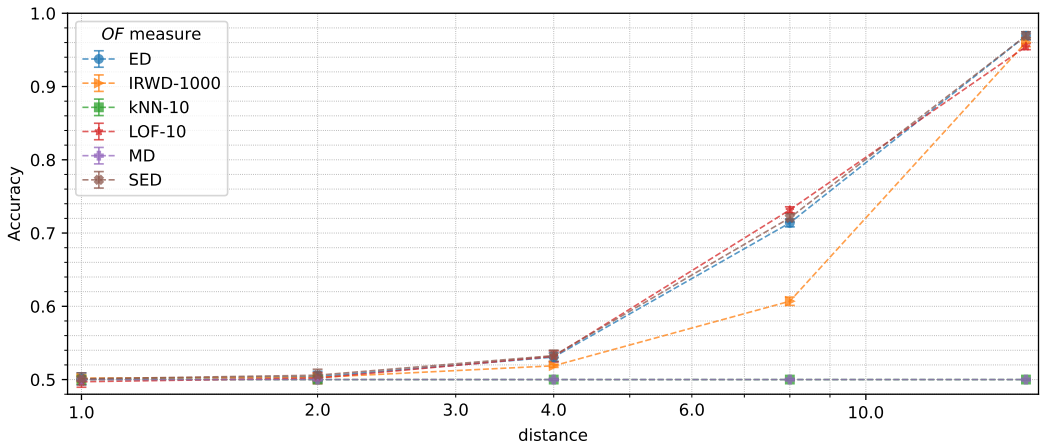


Figure 3.12: The performance of outlierness measures OF as affected by the number of training samples n. The fixed parameters in the experiment are: dimension of the feature space $d = 750$, distance to outliers $h = 8$ and distribution $G = Gaussian$. The results are aggregated for multiple generator seeds ξ and displayed as averages with error bars (standard deviation).

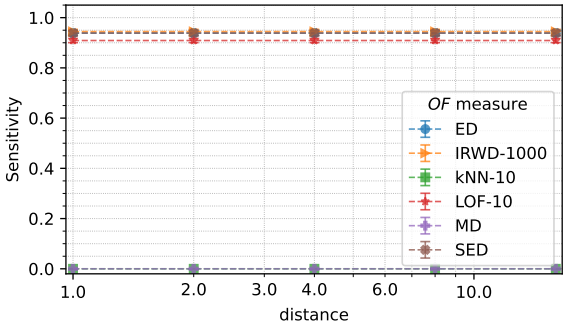
(a) Separability between in-distribution and out-of-distribution data



(b) Classification with respect to the training data



(c) Correctly recognized in-distribution



(d) Correctly recognized out-of-distribution

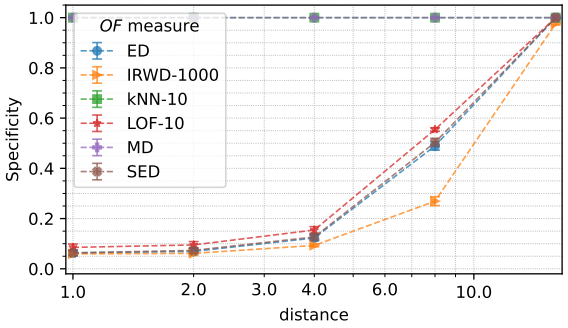


Figure 3.13: The performance of outlierness measures OF as affected by the distance to outliers h. The fixed parameters in the experiment are: dimension of the feature space $d = 750$, number of training samples $n = 2500$ and distribution $G = Gaussian$. The results are aggregated for multiple generator seeds ξ and displayed as averages with error bars (standard deviation).

3.2 Effect of feature correlations

The next experiment aims to analyze how the presence of correlations in data affects the performance of outlier detection methods. The research is conducted considering various number of features correlated and various correlation strength, under fixed number of training samples and dimension of the feature vectors.

3.2.1 Experiment organization

The experiment was organized similarly as the one described in the section 3.1.1.

The difference lies in the definitions of datasets T, K and U – utilizing only one generator G – the Multivariate Normal distribution (MVN). The number of training samples was fixed to $n = 2000$, as well as the dimension of feature vectors $d = 1000$. Instead, there are two new parameters introduced:

- the fraction of features that are correlated f_{corr},
- the strength of the features correlation g_{corr} (i.e., covariance value).

Both mentioned values affect the content of the covariance matrix Σ that is supplied to the MVN generator, e.g., for $d = 8$, $f_{corr} = 0.5$ and $g_{corr} = 0.25$ it would become

$$\Sigma_{ID} = \begin{bmatrix} \mathbf{1.00} & \mathbf{0.25} & \mathbf{0.25} & \mathbf{0.25} & 0.00 & 0.00 & 0.00 & 0.00 \\ \mathbf{0.25} & \mathbf{1.00} & \mathbf{0.25} & \mathbf{0.25} & 0.00 & 0.00 & 0.00 & 0.00 \\ \mathbf{0.25} & \mathbf{0.25} & \mathbf{1.00} & \mathbf{0.25} & 0.00 & 0.00 & 0.00 & 0.00 \\ \mathbf{0.25} & \mathbf{0.25} & \mathbf{0.25} & \mathbf{1.00} & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & \mathbf{1.00} & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & \mathbf{1.00} & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & \mathbf{1.00} & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & \mathbf{1.00} \end{bmatrix}. \tag{3.1}$$

By default the features are not correlated and the variance of all features equals 1.0.

Only the in-distribution (ID) data (sets T and K) are affected by the new parameters. The out-of-distribution (OOD) examples (set U) utilizes the identity matrix during generation (i.e., $\Sigma_{OOD} = \mathbb{I}$, features uncorrelated, variance equal to 1.0).

Summarizing, the input parameters that vary in the experiment are: the fraction of features that are correlated f_{corr}, the strength of the correlation g_{corr}, the distance to the outliers h and outlierness measure OF. The experiment was repeated several times with various values of the generator seed ξ.

3.2.2 Experiment results

Second experiment analyzes how the performance of outlierness measures is influenced by the correlations of features in dataset. Surprisingly, there are similarities noticeable when comparing the effect of changed correlation strength for a fixed number of correlated features (figure 3.14) and the effect of changed fraction of correlated features for a fixed variance (figure 3.15). The experiment involves $n = 2000$ training samples of dimension $d = 1000$ and distance to outliers $h = 8$.

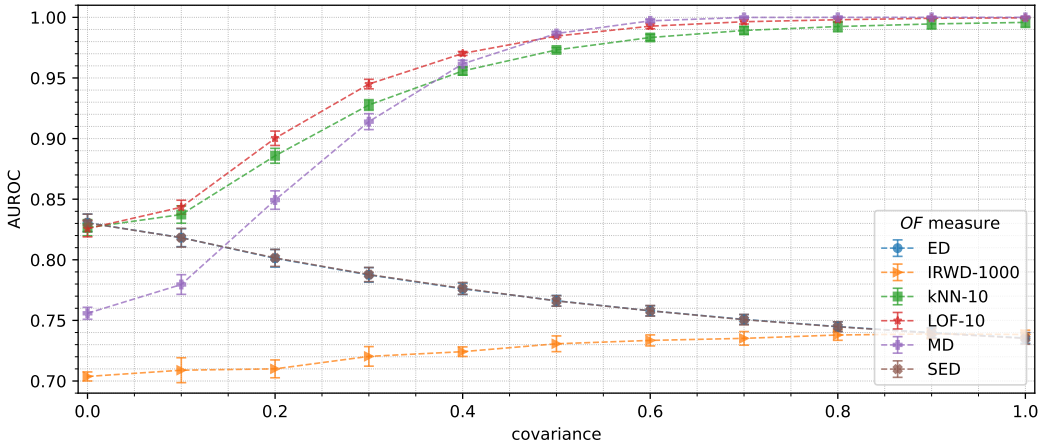
In particular, the ED and SED behave nearly identically in both cases. The correlation of features in data results in more concentrated distribution along a selected direction in the feature space, having an effect similar to the reduction of dimensionality. However, in case of measures that do not consider the correlations, like ED or SED, the result is visible as more extended distribution in space. This effect can be seen in figures 3.16d and 3.17d. The outcome is worse separability (AUROC value) and classification accuracy for more correlated features, which is caused by the scores for out-of-distribution data appearing within the distribution obtained for training data (especially visible in figure 3.17d).

The kNN, LOF and MD measures are capable of benefiting from the correlated features, however the nature behind this effect is different. The result in all cases is that the separability of the data is better (greater AUROC values); in general LOF performs slightly better than kNN in this experiment and MD is obtaining the third best separability in most cases for correlated data, i.e., for $f_{corr} > 0.1$ and/or $g_{corr} > 0.1$ (for uncorrelated data, ED and SED performs better than MD here). The effect on AUROC is similar regardless of whether the faction of correlated featured is increased or correlation strength (covariance value) is greater, but for MD the effect is slightly stronger for greater covariance value – i.e., for $g_{corr} \geq 0.4$ it outperforms kNN in the task of separability (and for $g_{corr} \geq 0.5$ it even outperforms LOF). Yet, both kNN and MD are suffering in the task of classification with respect to the training data – due to the same phenomenon already discussed in section 3.1, i.e., zero sensitivity, all testing in-distribution data are spuriously considered as outliers.

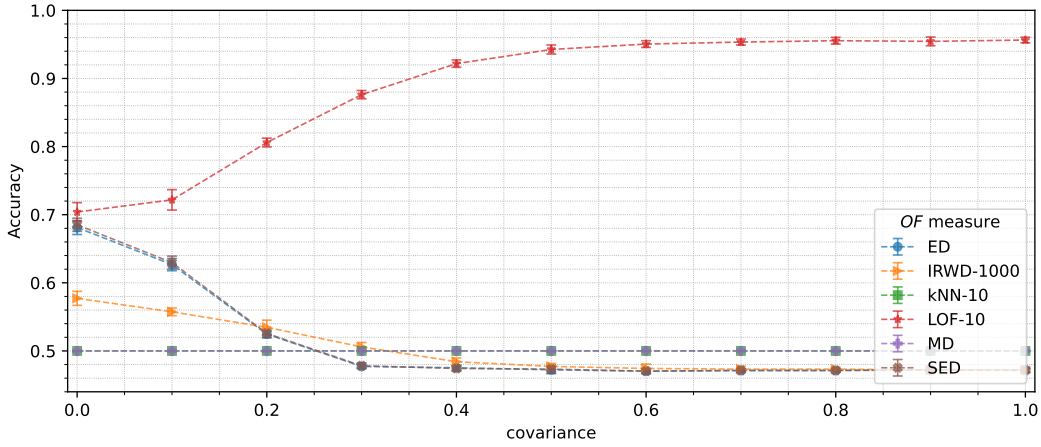
In conducted experiment, only the LOF maintains the good accuracy in the classification task, benefiting from the correlated features. Comparing the figures 3.16b and 3.17b, it is because out-of-distribution data are seen as more distant in case of stronger correlation in the in-distribution dataset.

Interesting observation can be made when comparing figures 3.16a and 3.17a (kNN) with figures 3.16c and 3.17c (MD). In both cases the better separation between in-distribution test data and out-of-distribution samples is visible for more correlated features. However, in case of kNN, the effect of data concentration is visible – obtained score values for training data and known examples (clusters T and K) are lower, corresponding with smaller spatial distances. Contrary, in case of MD, scores for both in-distribution data do not change significantly, yet still the outliers (cluster U) appears noticeably farther distanced.

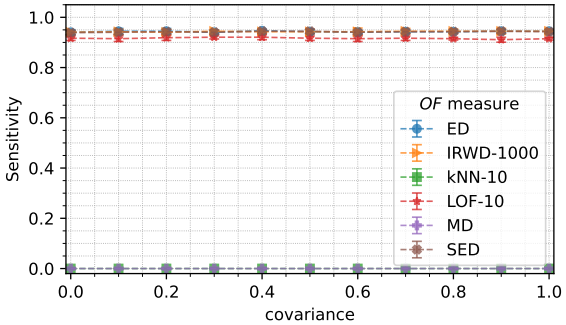
(a) Separability between in-distribution and out-of-distribution data



(b) Classification with respect to the training data



(c) Correctly recognized in-distribution



(d) Correctly recognized out-of-distribution

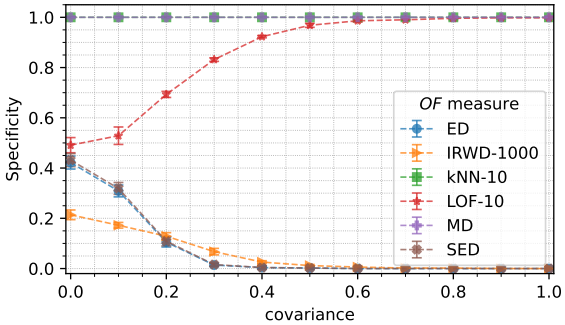
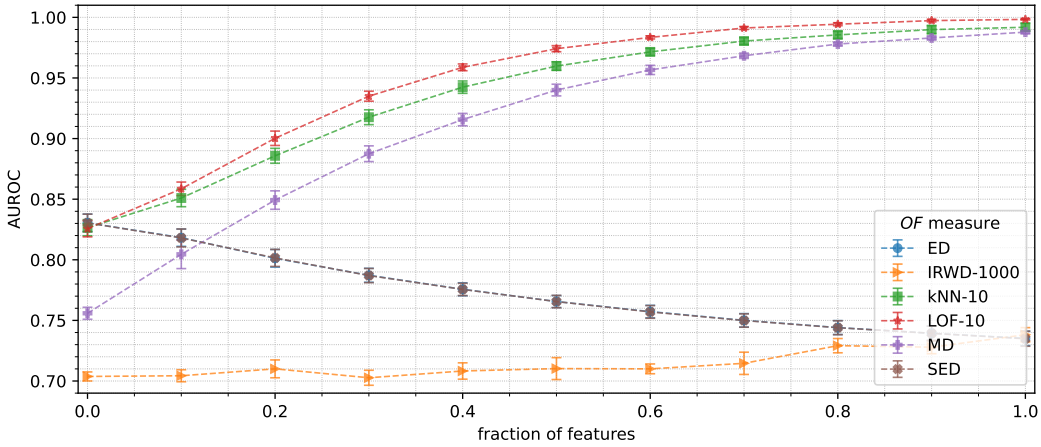
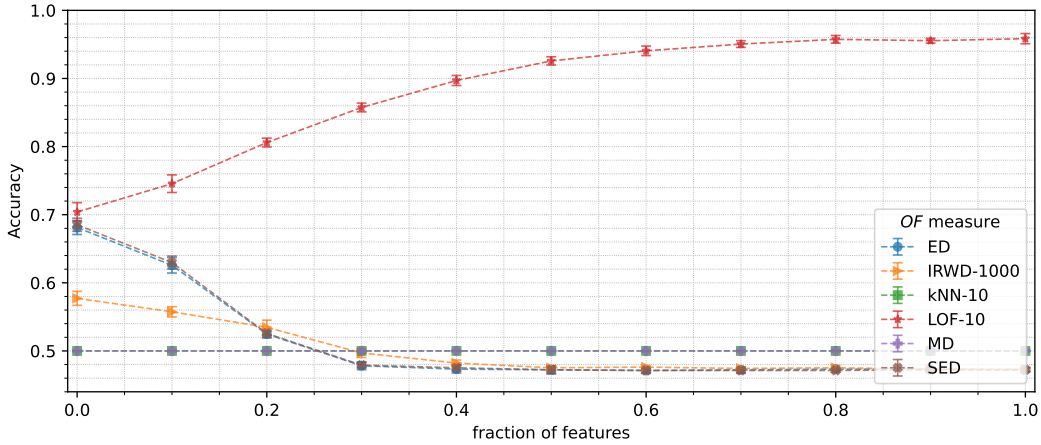


Figure 3.14: The performance of outlierness measures OF as affected by the correlation strength (covariance value g_{corr}). The fixed parameters in the experiment are: dimension of the feature space $d = 1000$, number of training samples $n = 2000$, distance to outliers $h = 8$ and fraction of features that are correlated $f_{corr} = 0.2$. The results are aggregated for multiple generator seeds ξ and displayed as averages with error bars (standard deviation).

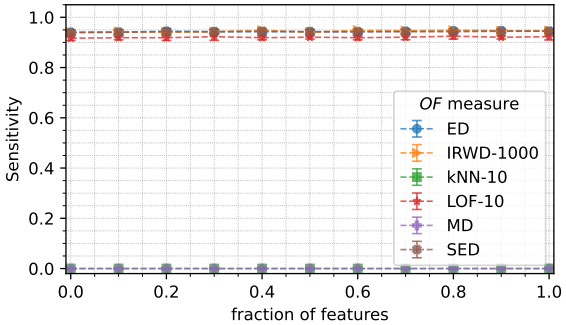
(a) Separability between in-distribution and out-of-distribution data



(b) Classification with respect to the training data



(c) Correctly recognized in-distribution



(d) Correctly recognized out-of-distribution

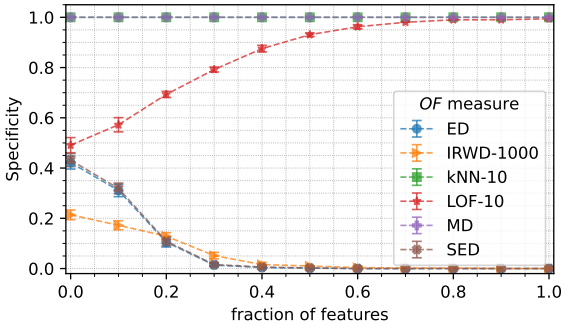
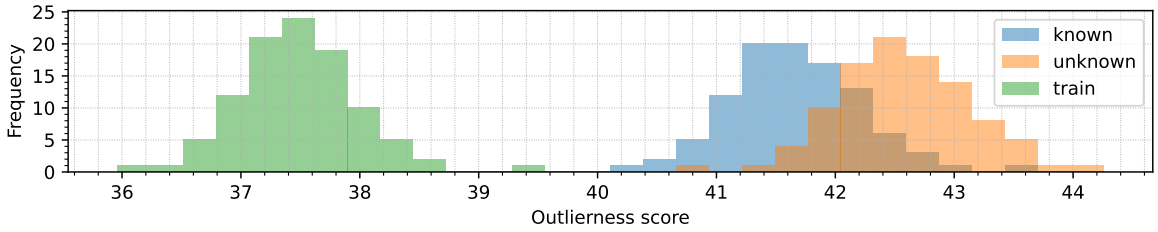
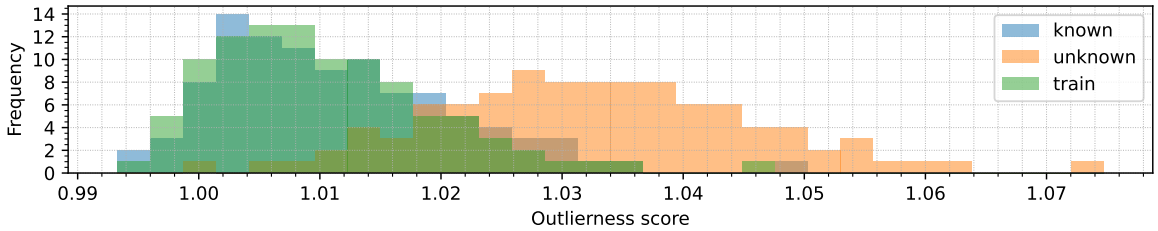


Figure 3.15: The performance of outlierness measures OF as affected by the fraction of features that are correlated f_{corr}. The fixed parameters in the experiment are: dimension of the feature space $d = 1000$, number of training samples $n = 2000$, distance to outliers $h = 8$ and correlation strength (covariance value $g_{corr} = 0.2$). The results are aggregated for multiple generator seeds ξ and displayed as averages with error bars (standard deviation).

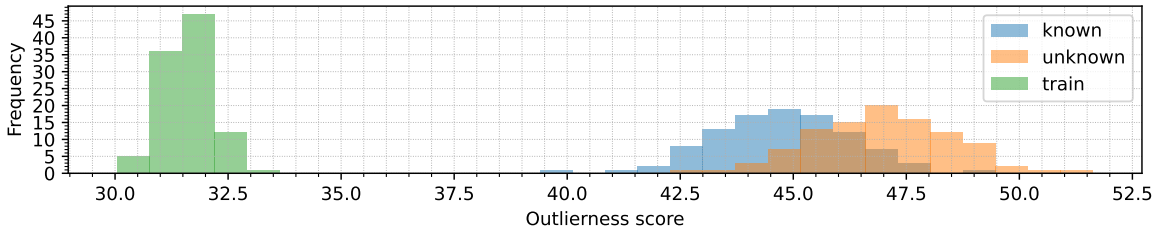
(a) k-Nearest Neighbors ($k = 10$), covariance: $g_{corr} = 0.2$, fraction: $f_{corr} = 0.2$



(b) Local Outlier Factor ($k = 10$), covariance: $g_{corr} = 0.2$, fraction: $f_{corr} = 0.2$



(c) Mahalanobis distance, covariance: $g_{corr} = 0.2$, fraction: $f_{corr} = 0.2$



(d) Standardized Euclidean distance, covariance: $g_{corr} = 0.2$, fraction: $f_{corr} = 0.2$

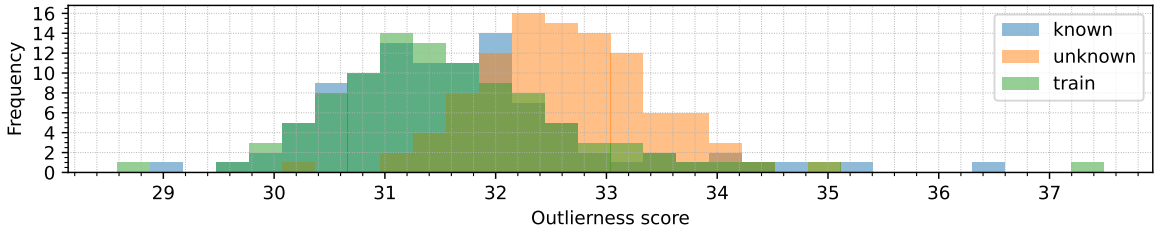
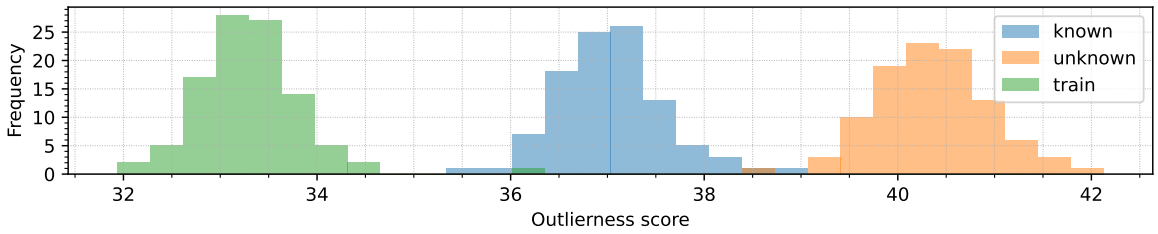
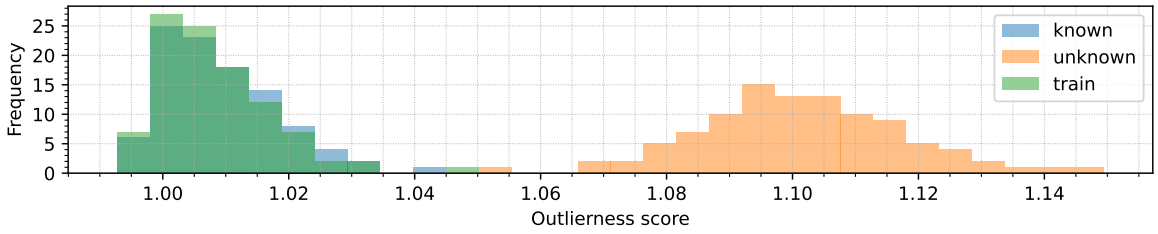


Figure 3.16: Distributions of outlierness scores for various measures OF with a small fraction of features slightly correlated. The correlation makes the generated clusters more concentrated in space, resulting in a better separability in case of kNN, LOF and MD, except for SED that is not considering correlations (compare with figure 3.17). Other experiment parameters involved: dimension of feature space $d = 1000$, number of training samples $n = 2000$, distance to outliers $h = 8$, generator seed $\xi = 0$.

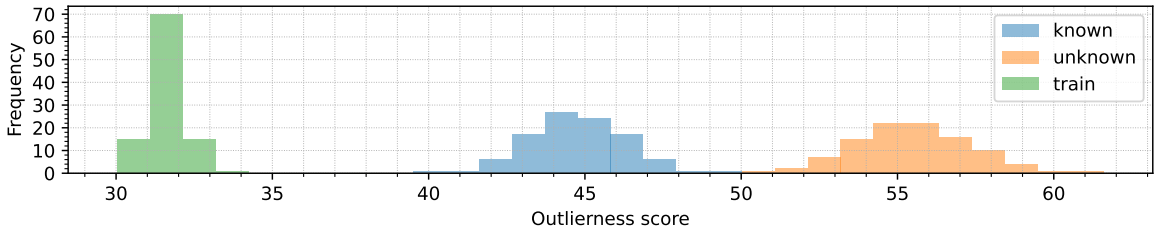
(a) k-Nearest Neighbors ($k = 10$), covariance: $g_{corr} = 0.5$, fraction: $f_{corr} = 0.5$



(b) Local Outlier Factor ($k = 10$), covariance: $g_{corr} = 0.5$, fraction: $f_{corr} = 0.5$



(c) Mahalanobis distance, covariance: $g_{corr} = 0.5$, fraction: $f_{corr} = 0.5$



(d) Standardized Euclidean distance, covariance: $g_{corr} = 0.5$, fraction: $f_{corr} = 0.5$

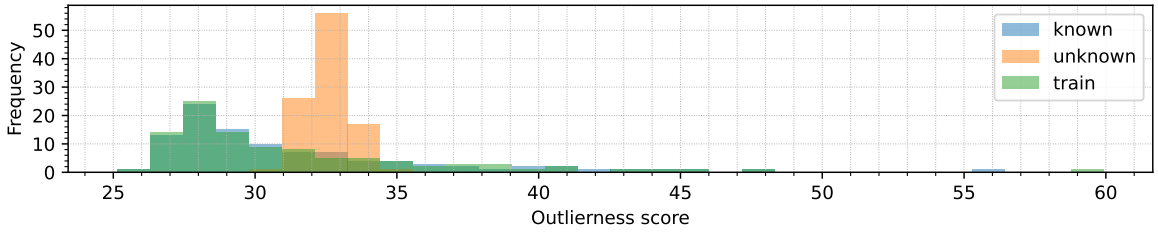


Figure 3.17: Distributions of outlierness scores for various measures OF with a half of features highly correlated. The significant correlation results in much better separability in case of kNN, LOF and MD; in case of not considering correlations SED measure, a long right tail appears (compare with figure 3.16). Other experiment parameters involved: dimension of feature space $d = 1000$, number of training samples $n = 2000$, distance to outliers $h = 8$, generator seed $\xi = 0$.

3.3 Influence of non-uniform variance of features

The third experiment is focused on analyzing the impact of the non-standardized features on the performance of the out-of-distribution detection techniques. The research considers such factors as the number of features with modified variance (other than default value: 1.0) and the modified variance's value itself. Similarly to previous experiment, the number of training samples and dimension of feature vectors are fixed.

3.3.1 Experiment organization

The experiment was organized similarly as the one described in the section 3.1.1.

The difference lies in the definitions of datasets T, K and U – utilizing only one generator G – the Multivariate Normal distribution (MVN). The number of training samples was fixed to $n = 2000$, as well as the dimension of feature vectors $d = 1000$. Instead, there are two new parameters introduced:

- the fraction of features that have modified (non-default) variance f_{var},
- the value of the modified variance g_{var}.

Both mentioned values affect the content of the covariance matrix Σ that is supplied to the MVN generator, e.g., for $d = 6$, $f_{var} = 0.5$ and $g_{var} = 0.25$ it would become

$$\Sigma_{ID} = \begin{bmatrix} \mathbf{0.25} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & \mathbf{0.25} & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & \mathbf{0.25} & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & \mathbf{1.00} & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & \mathbf{1.00} & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & \mathbf{1.00} \end{bmatrix}. \tag{3.2}$$

The features are not correlated here and the default variance of features equals 1.0.

Only the in-distribution (ID) data (sets T and K) are affected by the new parameters. The out-of-distribution (OOD) examples (set U) utilizes the identity matrix during generation (i.e., $\Sigma_{OOD} = \mathbb{I}$, features uncorrelated, variance equal to 1.0).

Summarizing, the input parameters that vary in the experiment are: the fraction of features with non-default variance f_{var}, the value of the non-default variance g_{var}, the distance to the outliers h and outlierness measure OF. The experiment was repeated several times with various values of the generator seed ξ.

3.3.2 Experiment results

Third experiment verifies how the outlierness measures perform when affected by the non-uniform variances of features in the dataset, i.e., part of the features in the in-distribution samples utilize a different variance.

Figure 3.18 illustrates how the variance value g_{var} influences the performance of outlierness measures. The custom variance value g_{var} is set on a fixed 20% of features ($f_{var} = 0.2$), while the remaining 80% features keep the default unitary variance (1.0). The experiment involves $n = 2000$ training samples of dimension $d = 1000$ and distance to outliers $h = 16$.

Cases where variance $g_{var} < 1.0$ correspond to concentrated distributions and therefore better spatial separation of data – perfect AUROC score (value 1.0) is obtained even for IRWD measure (left part of figure 3.18a), leading also to a higher classification accuracy (left part of figure 3.18b).

The results indicate significant decrease of separability where the custom variance $g_{var} > 1.0$ is applied (right part of figure 3.18a) – especially in case of measures that assume the same significance and values range of each feature: ED, IRWD, kNN and LOF. This situation corresponds to the values of selected 20% features from in-distribution data overlapping with values for the corresponding features from the out-of-distribution data. Relying on those features leads to losing the ability of proper outliers recognition (specificity) due to too wide spatial distribution of training data – the obtained distance scores for known in-distribution data and out-of-distribution samples start to overlap.

The MD and SED measures take into account the features variances and compensate their contribution to the spatial distance calculation, effectively distinguishing the in-distribution and out-of-distribution data based on the remaining 80% of features that do not overlap. In the conducted study, SED performs slightly better than MD in the separation task (figure 3.18a) and much better in the classification with respect to the training dataset (3.18b). The latter is due to the MD's lack of sensitivity, discussed already in the previous sections of this dissertation (mismatch between scores obtained for training T and testing K in-distribution samples).

It shall be noticed, that due to how the AUROC score calculation is performed[15], for variance value $g_{var} > 2.25$ (figure 3.18a) the unusual value of AUROC < 0.5 can be observed in case of ED, IRWD, kNN and LOF. This corresponds to the distributions histograms visible in the figures 3.20a, 3.20b and 3.21 – the outlierness scores of in-distribution data are greater (located to the right) than the scores obtained for out-of-distribution data. The interpretation of that case is: the out-of-distribution data are located closer to the training cluster T center (e.g. estimated mean) than the actual in-distribution samples. In some of such cases, the fine separability between the in-distribution data and the outliers can be made, such as in figures 3.21a or 3.21c. Notably, in cases visible in figures 3.21a and 3.21d, both ED and SED preserve outlierness scores for in-distribution data in the same range, so the very good classification accuracy could be reached if the classification criteria (section 2.2.3, formula 2.3) would have been revised accordingly.

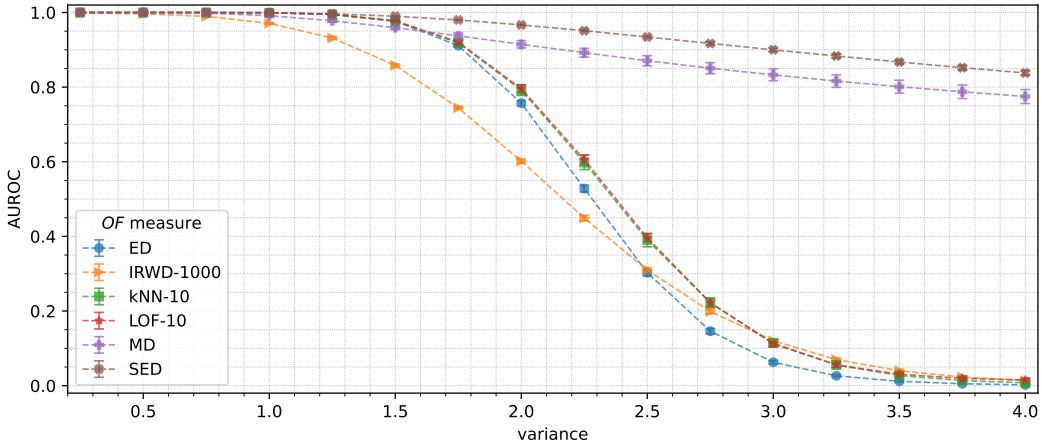
In some of the observed conditions (figures 3.20b and 3.21c) it is impossible to reliably calibrate kNN and MD for classification, because the scores obtained for out-of-distribution data are located between scores observed for training and testing in-distribution data.

Figure 3.19 shows the performance of outlierness measures when a fraction of features f_{var} contains a custom, fixed variance. The fixed variance value is $g_{var} = 1.5$; the remaining fraction of features maintains the default unitary variance (1.0). Like in previous example, the results are shown for case involving $n = 2000$ training samples of dimension $d = 1000$ and distance to outliers $h = 16$. The results in this case are analogous the previously discussed case, however now the parameter effect bears stronger impact on all the analyzed measures. Still, both MD and SED perform better than other measures in terms of data separation potential (figure 3.19a), as long as the fraction of features with custom variance does not bring the outliers too close to the training data. For $f_{var} = 0.6$ the in-distribution data are non-distinguishable from the outliers for MD and SED; for ED, kNN and LOF the same effect was observed for $f_{var} = 0.5$.

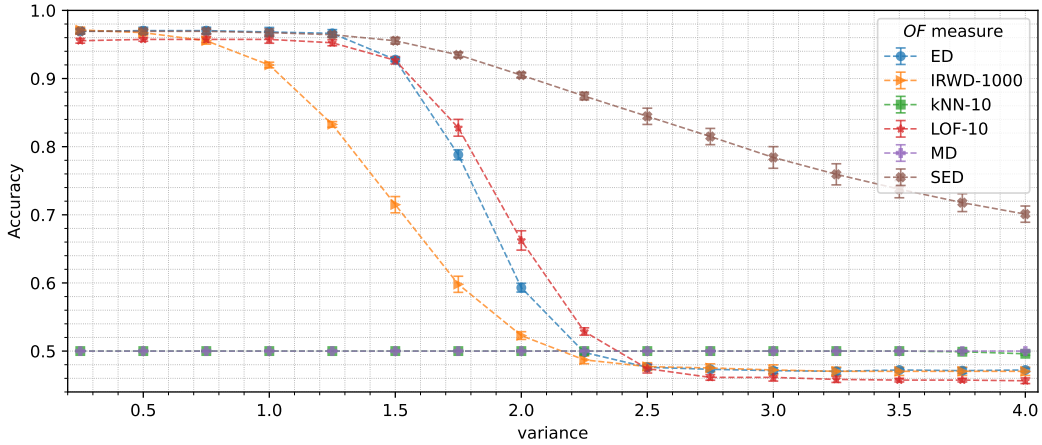
Overall, the experiment suggests that the analysis of variances and the standardization of data may be necessary to maintain optimal performance of methods such as IRWD, kNN and LOF, while measures like MD or SED work out of the box in the considered scenarios.

[15]Implementation detail: the `roc_curve()` from scikit-learn library [51] assumes higher scores to be associated with a positive label, i.e., treating the scores like the class probability value returned by the classifier; if the scores have different interpretation, like in the conducted study (distance - lower scores indicate positive class), the function parameters have to be adjusted, e.g., by inverting the labels; `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html`

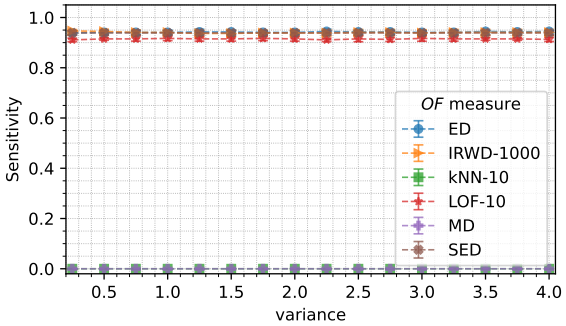
(a) Separability between in-distribution and out-of-distribution data



(b) Classification with respect to the training data



(c) Correctly recognized in-distribution



(d) Correctly recognized out-of-distribution

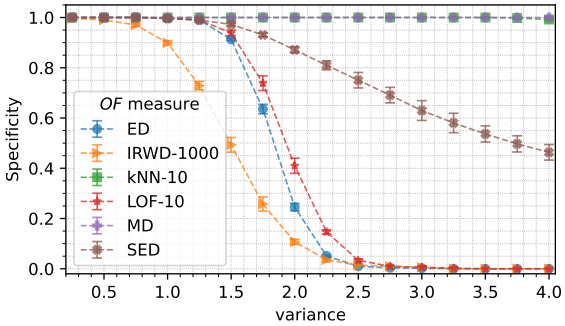
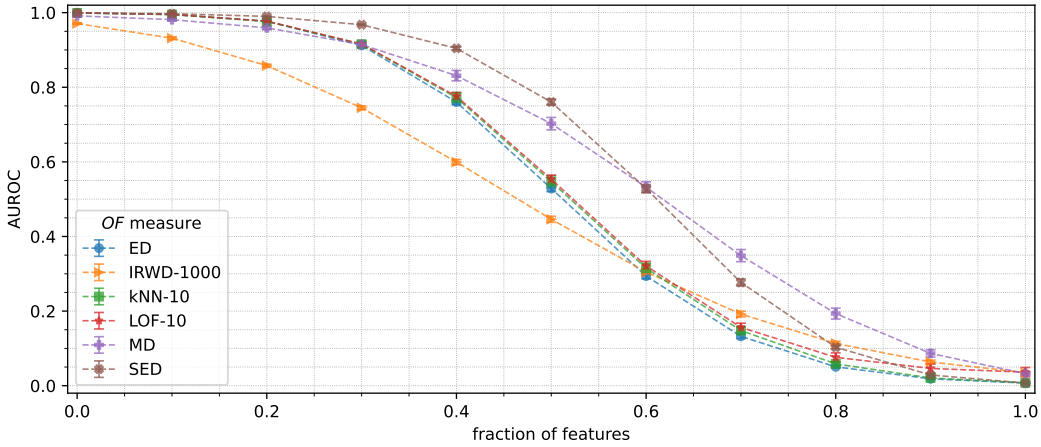
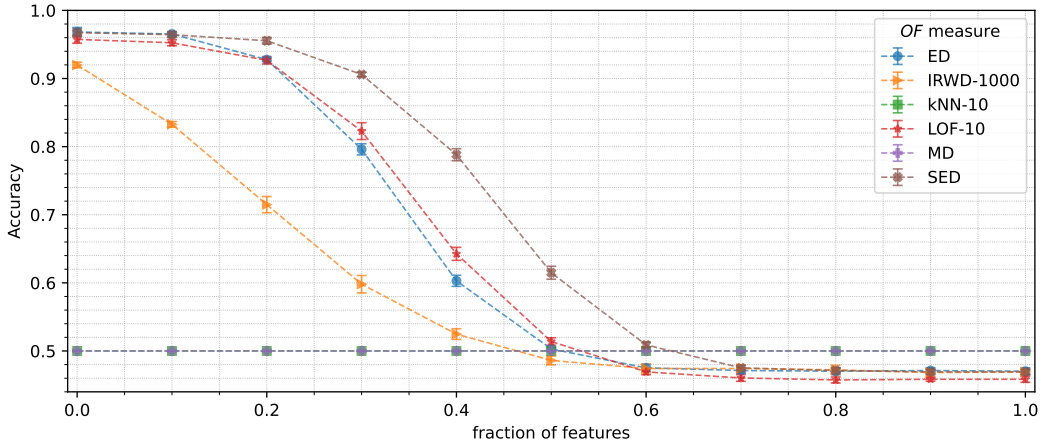


Figure 3.18: The performance of outlierness measures OF as affected by variance of features value g_{var}. The fixed parameters in the experiment are: dimension of the feature space $d = 1000$, number of training samples $n = 2000$, distance to outliers $h = 16$ and fraction of features that have modified variance $f_{var} = 0.2$. The results are aggregated for multiple generator seeds ξ and displayed as averages with error bars (standard deviation).

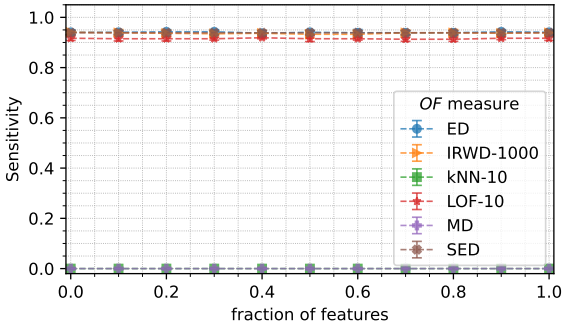
(a) Separability between in-distribution and out-of-distribution data



(b) Classification with respect to the training data



(c) Correctly recognized in-distribution



(d) Correctly recognized out-of-distribution

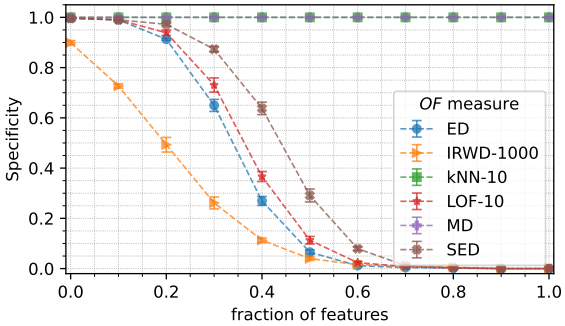
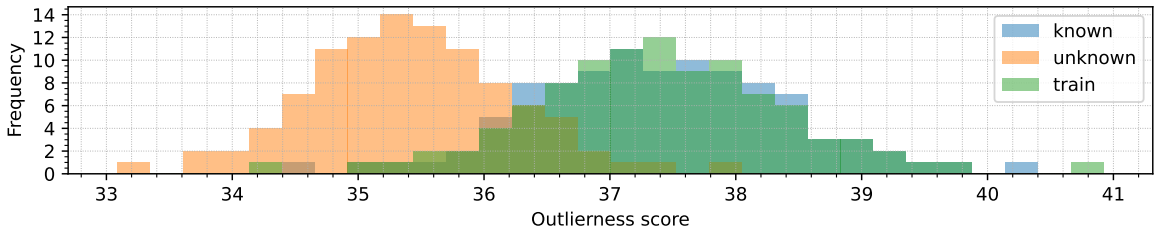
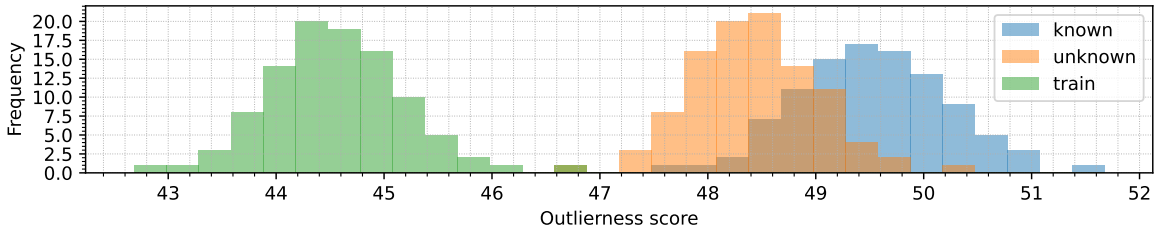


Figure 3.19: The performance of outlierness measures OF as affected by the fraction of features that have modified variance f_{var}. The fixed parameters in the experiment are: dimension of the feature space $d = 1000$, number of training samples $n = 2000$, distance to outliers $h = 16$ and variance of features value $g_{var} = 1.5$. The results are aggregated for multiple generator seeds ξ and displayed as averages with error bars (standard deviation).

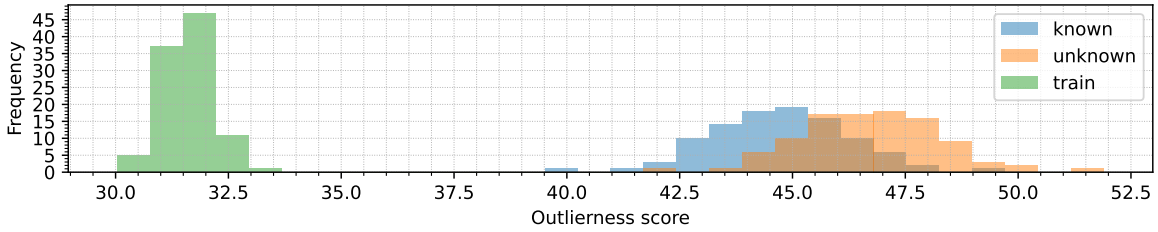
(a) Euclidean distance, variance: $g_{var} = 3.0$, fraction: $f_{var} = 0.2$



(b) k-Nearest Neighbors ($k = 10$), variance: $g_{var} = 3.0$, fraction: $f_{var} = 0.2$



(c) Mahalanobis distance, variance: $g_{var} = 3.0$, fraction: $f_{var} = 0.2$



(d) Standardized Euclidean distance, variance: $g_{var} = 3.0$, fraction: $f_{var} = 0.2$

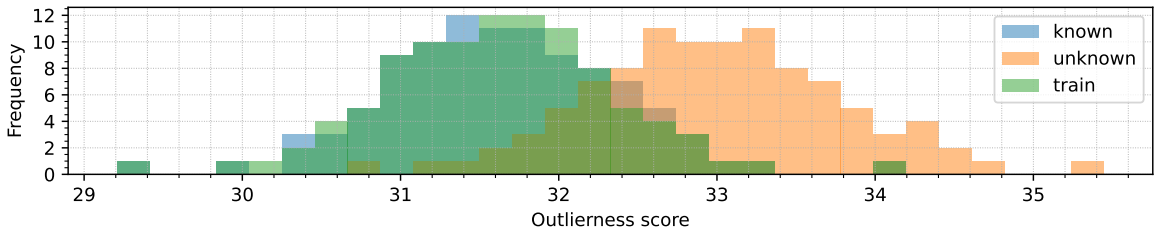
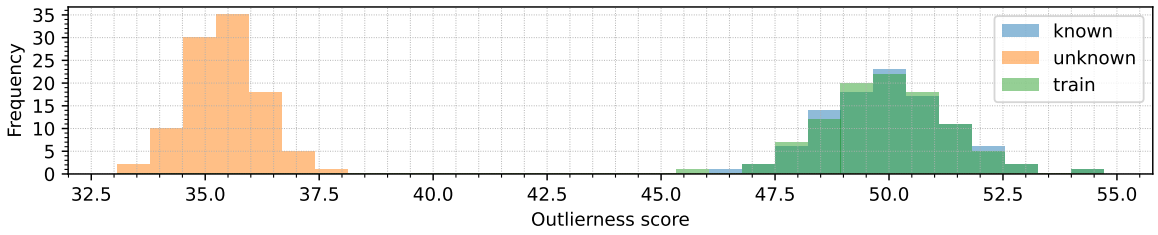
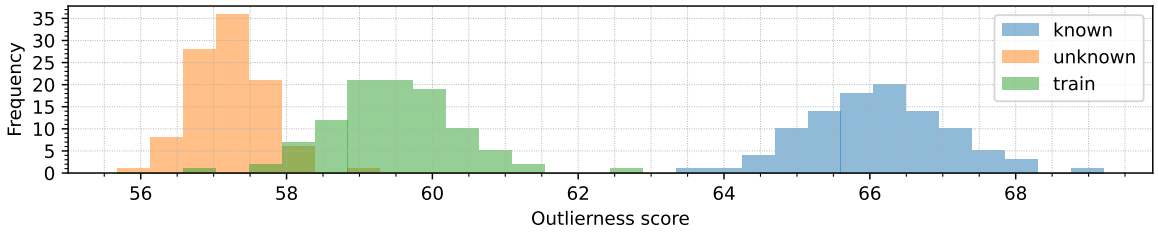


Figure 3.20: Histograms of outlierness scores for various measures OF considering data distribution with a small fraction of features having bigger variance. Due to bigger variance the outliers starts to overlap with in-distribution data in space, yet measures that consider the variance (MD, SED) maintain the ability to separate the clusters, while for kNN the outliers appear closer to training data than the known in-distribution examples. Other experiment parameters involved: dimension of feature space $d = 1000$, number of training samples $n = 2000$, distance to outliers $h = 16$, generator seed $\xi = 0$.

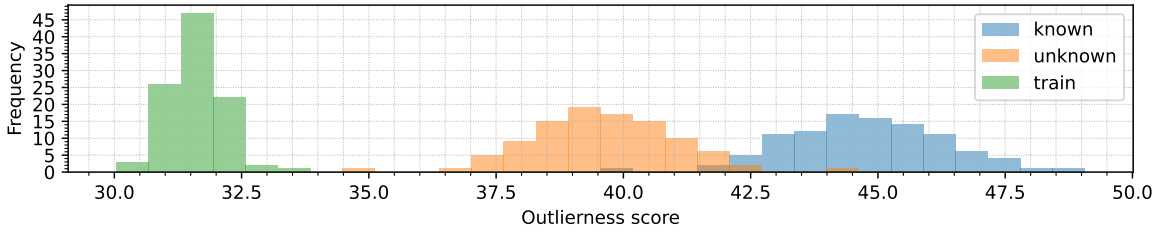
(a) Euclidean distance, variance: $g_{var} = 4.0$, fraction: $f_{var} = 0.5$



(b) k-Nearest Neighbors ($k = 10$), variance: $g_{var} = 4.0$, fraction: $f_{var} = 0.5$



(c) Mahalanobis distance, variance: $g_{var} = 4.0$, fraction: $f_{var} = 0.5$



(d) Standardized Euclidean distance, variance: $g_{var} = 4.0$, fraction: $f_{var} = 0.5$

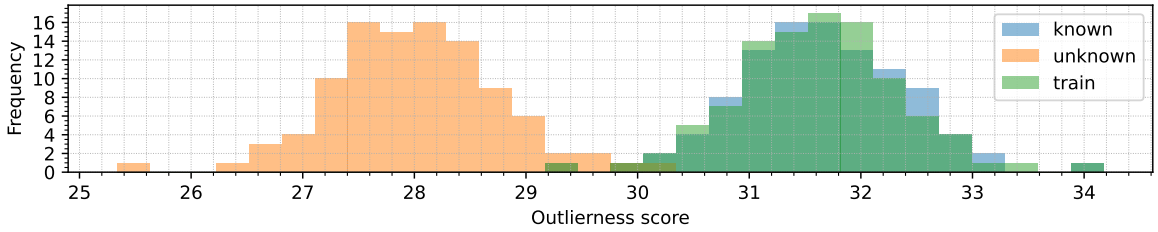


Figure 3.21: Histograms of outlierness scores for various measures OF considering data distribution with a significant fraction of features having very big variance. In such conditions it is impossible to reliably calibrate kNN and MD for classification, while both ED and SED preserve scores for in-distribution data in the same range (however, the formula 2.3 would have to be revised for classification). Other experiment parameters involved: dimension of feature space $d = 1000$, number of training samples $n = 2000$, distance to outliers $h = 16$, generator seed $\xi = 0$.

3.4 Overlapping and accurate representations

The goal of the next experiment is to identify the required conditions for maintaining the matching representations of in-distribution samples, i.e., overlapping histograms for training and testing data presented in previous sections (e.g., figures 3.4, 3.5 and 3.6). The research is conducted, considering primarily the dimensions of feature vectors d and the number of samples n.

3.4.1 Experiment organization

The experiment is organized as follows:

- First, 2 data clusters are generated.
  - Both datasets (K_1, K_2) are produced from the same chosen generator G (*Gaussian* [uncorrelated], *MVN* [with part of features correlated], *triangular* or *uniform* distribution – that is located around the center of the coordinate system $[0, 0, \ldots, 0]$ with spread of ± 1), each dataset containing n samples of dimension d.
- Then the bounding boxes around the clusters K_1 and K_2 are constructed.
- Next the common part of the bounding boxes for K_1 and K_2 is identified.
- The volumes of discovered boxes are calculated: V_{K_1}, V_{K_2}, $V_{K_1 \cap K_2}$.
  - The boxes are hyperrectangles in $\mathbb{R}^d$ space, with volumes defined as

$$V_K = \prod_{j=1}^{d} l_j \ , \tag{3.3}$$

    where l_j is the length of the box's edge that is parallel to the j-th axis,

$$l_j = \max\Big\{K[*, j]\Big\} - \min\Big\{K[*, j]\Big\}. \tag{3.4}$$

  - For high dimensions, e.g., $d \geq 1000$, the calculated volumes can become so huge that the values hit the overflow in memory (i.e., $\pm\infty$ in the Floating-Point Arithmetic [IEEE-754] [48][23]). Hence, in computation, for efficiency the logarithmic representation was involved,

$$Vl_K = log_{10}(V_K) = \sum_{j=1}^{d} log_{10}(l_j) \ , \tag{3.5}$$

where the result represent the order of magnitude, e.g., $Vl_K = 3$ means the volume is equal to 1000 and $Vl_K = 21$ corresponds to a volume of 10^{21} units.

- Finally the Jaccard index of the bounding boxes is calculated – also known as the Intersection over Union metric, defined as the ratio between the overlapping area/volume and the union of two areas/volumes,

$$J(K_1, K_2) = IoU(K_1, K_2) = \frac{V_{K_1 \cap K_2}}{V_{K_1 \cup K_2}} = \frac{V_{K_1 \cap K_2}}{V_{K_1} + V_{K_2} - V_{K_1 \cap K_2}}. \tag{3.6}$$

Summarizing, the input parameters varying in the experiment are: number of samples n, dimension of the feature space d and the generator distribution G. The experiment was repeated several times with various values of the generator seed ξ (that affected the values within K_1 and K_2) to observe the variability of results.

3.4.2 Experiment results

The fourth experiment explores the properties of the high-dimensional spaces, analyzing the requirements for obtaining the accurate representations of the data clusters represented with the bounding boxes and previously discussed distance measures (section 2.3).

Figure 3.22 displays the non-intuitive phenomenon of difficulty related to obtaining matching (overlapping) representations of two data clusters produced by the same distribution. The idea is to construct bounding boxes around the generated clusters, i.e., identifying minimum and maximum values of each feature (vector component) and identify the common part (intersection) of those two boxes. Then, the ratio between the volume of identified common part and the total volume covered by the two boxes is calculated – so called Jaccard index (formula 3.6).

It turns out, that obtaining accurate representations that way requires a significant number of samples n and becomes especially difficult in higher dimension d. In addition, while it is possible to accomplish this for the distributions generators with finite output domain ($G = Triangular$, $G = Uniform$), it becomes gradually challenging for the $G = Gaussian$ distribution ($G = MVN$ in general) even in dimensions like $d \sim 10$. Surprisingly, involving the correlations of features in the generator does not seem to influence this task significantly (compare figures 3.23a and 3.23b), although in practice it results in more concentrated clusters with smaller distances between points.

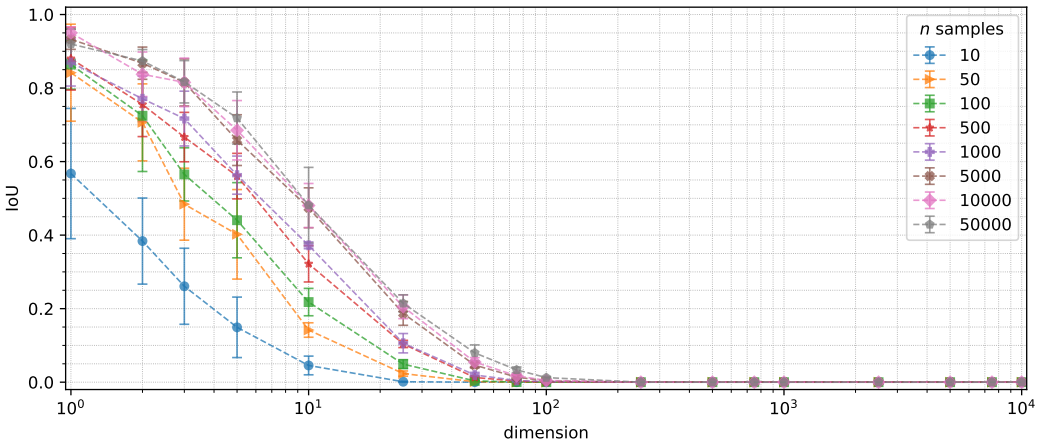
Figure 3.23c presents a different attempt for obtaining a cluster representation, based on the training and testing data (clusters T and K) from experiment described in section 3.1. In this approach, the clusters are represented using the Mahalanobis distance model, assuming hyperellipsoid-like structure of data, estimated using the covariance matrix and distances of the clusters points with respect to the ellipsoid center (mean values of features). Instead of Jaccard index, the sensitivity (True Positive Rate) is reported – counting the point from the testing cluster as from the same distribution, if it not farther than 99% of the training data.

Utilizing this approach, achieving the accurate representation of the clusters is possible when providing sufficiently large number of samples n for a given dimension d of feature space. Surprisingly, the observed relation is very similar to the bounding boxes estimation obtained for $G = Uniform$ distribution (compare 3.22c and 3.23c). Note that reaching the accurate representation requires significantly more samples n than the bare minimum required for estimating the covariance matrix in dimension d – minimum: $n \geq d$; optimum: $n \gtrsim 50 \cdot d$ (as for analyzed range in figure 3.23c). This observation corresponds the phenomenon presented previously in figure 3.4 and justifies the utilization of the pooled covariance matrix when calculating Mahalanobis distance (section 2.3.6, formula 2.28), as obtaining more samples can lead to more accurate representation of the training cluster and higher classifier sensitivity.

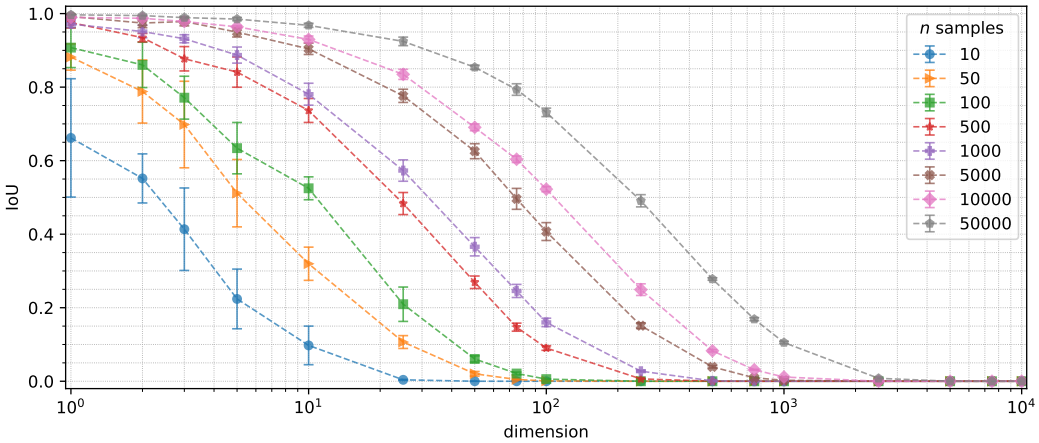
Figure 3.24 presents comparison of other attempts of obtaining a cluster representation using the distance measures described in section 2.3. Figure 3.24a illustrates the previously discussed phenomenon of kNN measure mismatching between training and testing samples coming from the same distribution, depending on the chosen parameter k value, as visible also in figure 3.5. In case of $k = 10$, any number of samples $n \geq 500$ does not improve the sensitivity (True Positive Rate). For greater k values, the overall trend is preserved but appears shifted to the right in the figure, e.g., reaching $TPR \approx 0.9$ for $k = 20$ and $d = 100$. This observation deserves a dedicated, separate further study.

Interestingly, the LOF measure appears nearly unaffected by dimensionality of the feature space in this task, reaching good score for all but extreme case ($n = 10$). The SED measure shows behavior similar to MD for the analyzed dimensions d range, however the number of samples n necessary to obtain accurate representation is much lower than for MD. This corresponds with the results observed in figures 3.7, 3.9c and 3.9d in subsection 3.1.2.

(a) Distribution $G = Gaussian$



(b) Distribution $G = Triangular$



(c) Distribution $G = Uniform$

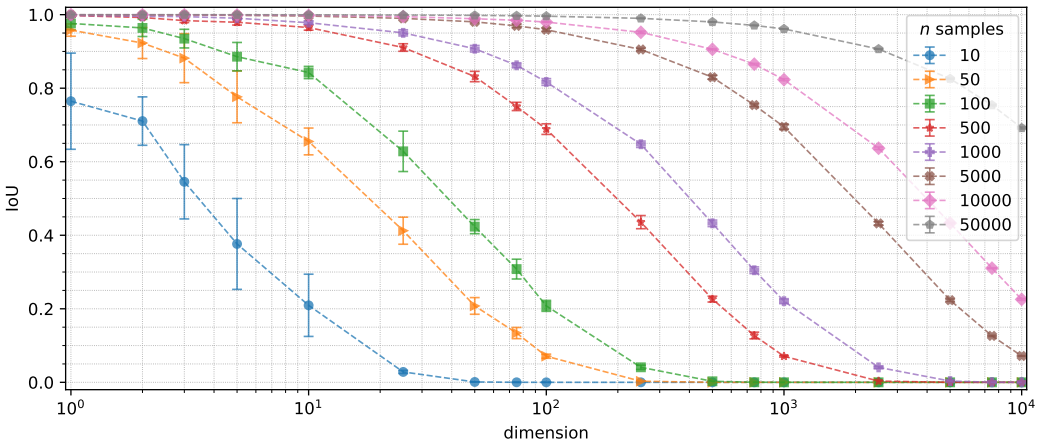
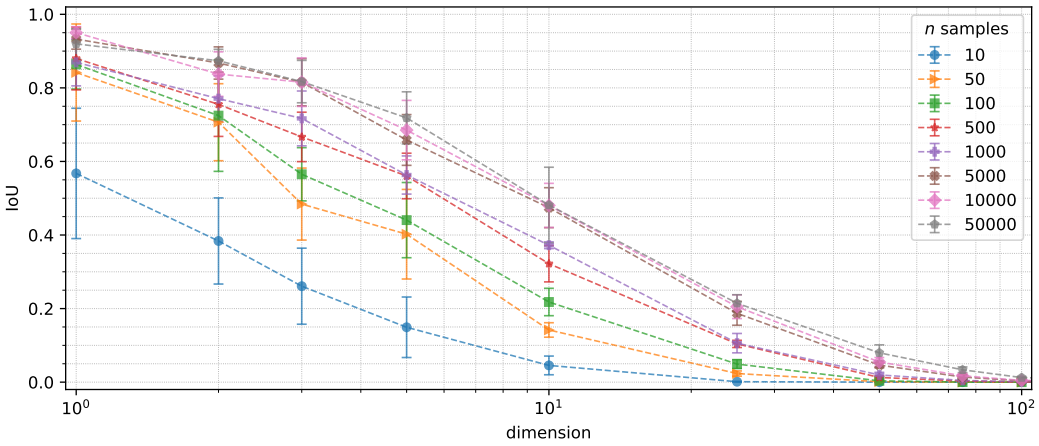
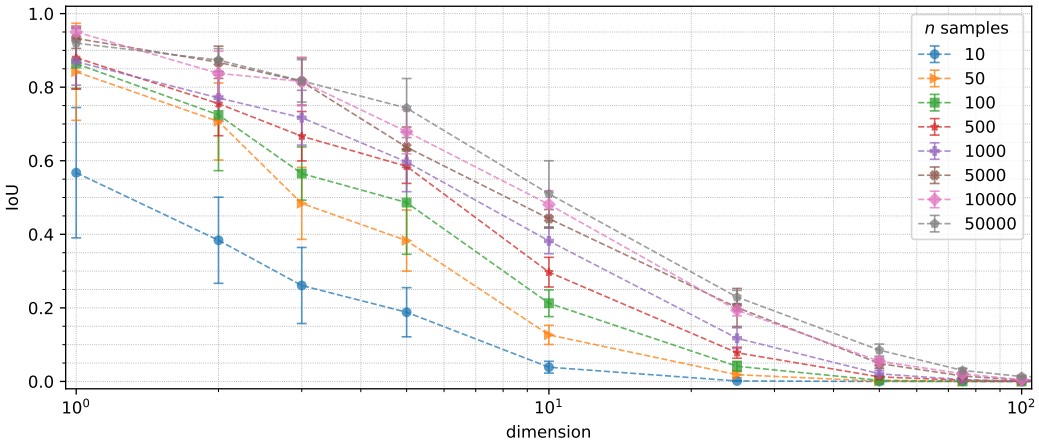


Figure 3.22: The relation between number of samples n and dimension of feature space d in the task of recreating the same data cluster (represented as the bounding box). This task turns out significantly more difficult for $G = Gaussian$ distribution than for the distributions with finite output domain ($G = Triangular$, $G = Uniform$). The results are aggregated for multiple generator seeds ξ and displayed as averages with error bars (standard deviation).

(a) Distribution $G = Gaussian$ (detailed view)



(b) Distribution $G = MVN$ ($g_{corr} = 0.5$, $f_{corr} = 0.5$)



(c) Distribution $G = Gaussian$, utilizing Mahalanobis distance

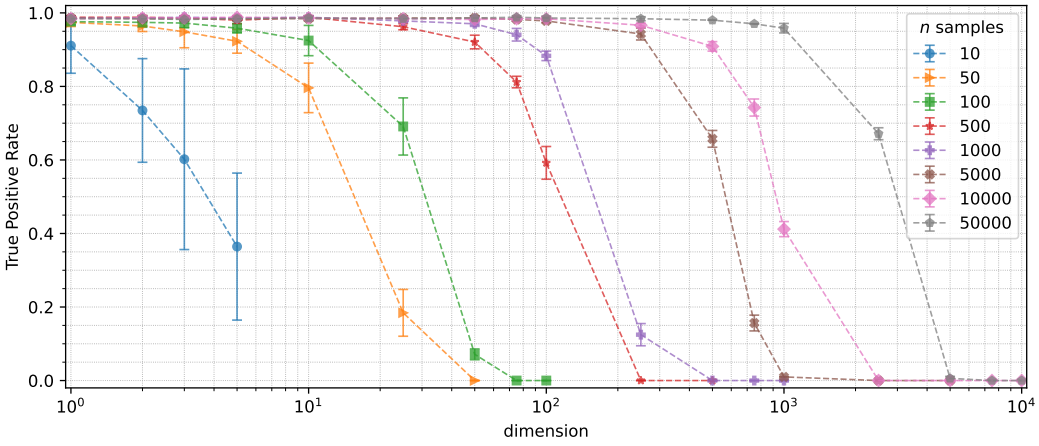
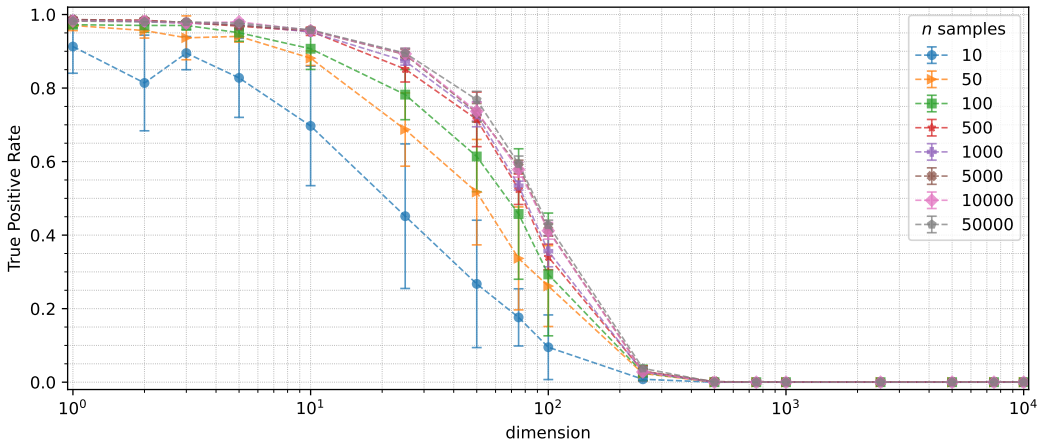
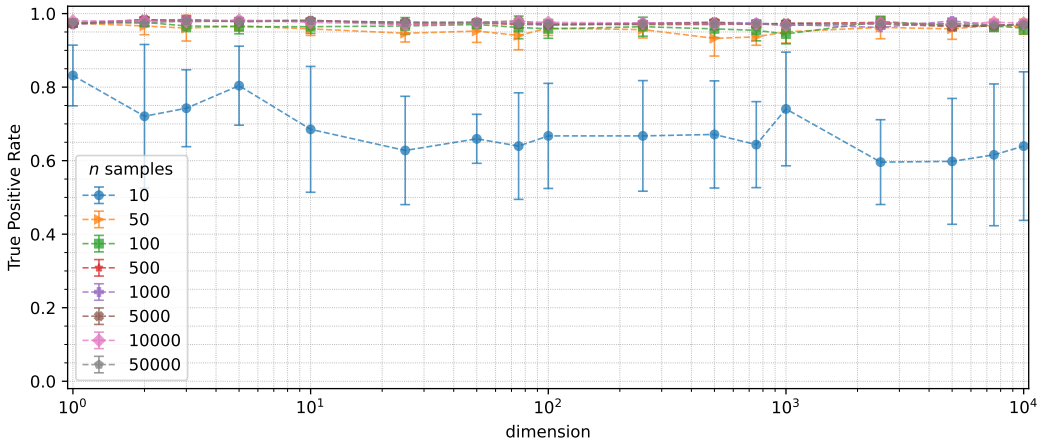


Figure 3.23: The correlation of features, despite resulting in more concentrated clusters (smaller distances), is does not affect the results significantly (comparing subfigures 3.23a and 3.23b). However, representing cluster with Mahalanobis distance model (i.e., hyperellipsoid-like structure), the overlapping can be effectively achieved ever for higher dimensions d. The results are aggregated for multiple generator seeds ξ and displayed as averages with error bars (standard deviation).

(a) Distribution $G = Gaussian$, utilizing k-Nearest Neighbors ($k = 10$)



(b) Distribution $G = Gaussian$, utilizing Local Outlier Factor ($k = 10$)



(c) Distribution $G = Gaussian$, utilizing Standardized Euclidean distance

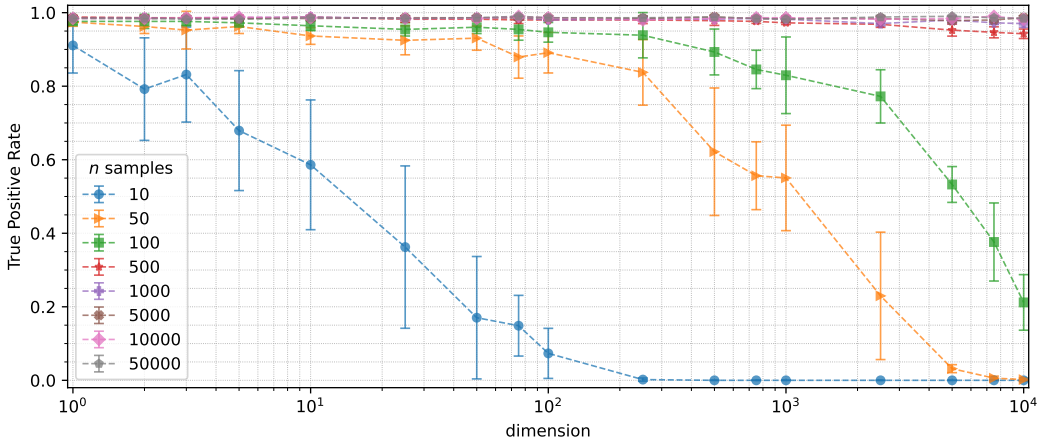


Figure 3.24: Clusters representations based on the distance measures described in section 2.3 allow to obtain the effective overlapping ever for high features space dimensions d. For some methods the number of samples n does not need to be high to perform well; kNN's performance is related it selected parameter k value. The results are aggregated for multiple generator seeds ξ and displayed as averages with error bars (standard deviation).

3.5 Parameter estimation errors

The goal of the last numerical experiment is to explain the reason behind low sensitivity of MD, especially when compared with conceptually similar SED measure, e.g., visible when comparing the histograms in figures 3.16c and 3.16d, also observing the rapid sensitivity decay for MD in higher dimensions d in figure 3.11c. Both mentioned measures rely on estimating the vector of means μ and either the full covariance matrix Σ or only its diagonal elements (variances). The analysis of errors impact is conducted as a function of the feature space dimension d and the number of samples n.

3.5.1 Experiment organization

The experiment is organized as follows:

- First, a single data cluster K is generated.
  - The dataset is produced from the Multivariate Normal distribution (MVN), containing n samples of dimension d, located around the center of the coordinate system $\mu = [0, 0, \ldots, 0]$ with a spread of ± 1).
  - The distribution shape is influenced by two additional parameters:
    - the fraction of features that are correlated f_{corr},
    - the strength of the features correlation g_{corr}.

    These parameters affect the content of the covariance matrix Σ supplied to the MVN generator, e.g., for $d = 5$, $f_{corr} = 0.6$ and $g_{corr} = 0.25$ it results in

$$\Sigma = \begin{bmatrix} \mathbf{1.00} & \mathbf{0.25} & \mathbf{0.25} & 0.00 & 0.00 \\ \mathbf{0.25} & \mathbf{1.00} & \mathbf{0.25} & 0.00 & 0.00 \\ \mathbf{0.25} & \mathbf{0.25} & \mathbf{1.00} & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & \mathbf{1.00} & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & \mathbf{1.00} \end{bmatrix}. \tag{3.7}$$

- The sample mean $\hat{\mu}$ and empirical covariance $\hat{\Sigma}$ are estimated on the dataset K.
- The Mean Squared Error (MSE) of the estimates are calculated as per formula,

$$MSE(A, \hat{A}) = \frac{1}{w} \cdot \sum_{i=1}^{w} \left(A_i - \hat{A}_i \right)^2, \tag{3.8}$$

where A is the true/expected value of a parameter and $\hat{A}$ is its estimated value, w is the number of components if A is a vector or matrix, i indexes the components. In particular, the inaccuracies of the following parameters were computed:

- means: $MSE(\mu, \hat{\mu})$; $w = d$;
- covariance matrices: $MSE(\Sigma, \hat{\Sigma})$; $w = d^2$;
- variances: $MSE(\Sigma_{i,j}, \hat{\Sigma}_{i,j})$ where $\forall(i,j) \in \{1, 2, \ldots, d\} : i = j$; $w = d$;
- correlations: $MSE(\Sigma_{i,j}, \hat{\Sigma}_{i,j})$ where $\forall(i,j) \in \{1, 2, \ldots, d\} : i \neq j$; $w = \frac{d}{d-1}$.

Summarizing, the input parameters that vary in the experiment are: the number of samples n, dimension of the feature space d, the fraction of features that are correlated f_{corr} and the strength of the correlation g_{corr}. The experiment was repeated several times with various values of the generator seed ξ.

3.5.2 Experiment results

The fifth experiment is a numerical study of how accurate the empirical estimations of distribution properties are, considering number of samples in cluster n and the dimension d of vectors, focusing in the high-dimensional feature spaces in particular.

Figure 3.25 presents the results obtained for a Multi-variate Normal (MVN) distribution with small fraction of features $f_{corr} = 0.2$ slightly correlated $g_{corr} = 0.2$. The experiment is conducted for a range of correlation settings and the results do not differ significantly – unless a very strong correlation of features in data is involved (as shown in the figure 3.26.

Surprisingly, all three analyzed parameters: means, variances and covariances; are estimated with similar accuracy for a given experiment configuration (n, d, f_{corr}, g_{corr}). The estimation errors are primarily related to the number of available data samples n, with the behavior being inversely proportional – the more examples provided, the lesser error (more accurate representation of the original/expected values). The estimation accuracy appears not significantly affected by the dimensionality of the feature space.

Comparing the results side-by-side (figures 3.25a, 3.25b and 3.25c) it can be noticed that the estimation errors for means and covariances appear nearly the same, especially for higher dimensionality ($d \geq 100$), while the estimation of variances appear slightly worse (higher error). However, all estimation errors for a given set of parameters (n, d, f_{corr}, g_{corr}) are of the same order of magnitude.

- $d = 1000$, $f_{corr} = 0.2$, $g_{corr} = 0.2$, $n = 1000$:
  - $MSE(\text{means}) \approx 1.0 \cdot 10^{-3}$,
  - $MSE(\text{variances}) \approx 0.9 \cdot 10^{-3}$,
  - $MSE(\text{covariances}) \approx 1.0 \cdot 10^{-3}$;
- $d = 1000$, $f_{corr} = 0.2$, $g_{corr} = 0.2$, $n = 10000$:
  - $MSE(\text{means}) \approx 1.0 \cdot 10^{-4}$,
  - $MSE(\text{variances}) \approx 0.9 \cdot 10^{-4}$,
  - $MSE(\text{covariances}) \approx 1.0 \cdot 10^{-4}$.

This observation is significant, because when considering how the computed values are involved in the distances calculation for the MD measure (section 2.3.6, formula 2.24) or the SED measure (section 2.3.7, formula 2.29), the various cumulative error can be estimated. Assuming the average estimation error of a single parameter, e.g., the variance on the j-th axis or the covariance between features j_1 and j_2 (element Σ_{j_1,j_2}), being δ in both cases (same order of magnitude), when computing the Standardized Euclidean distance, each of the d estimated variances scales the corresponding distances (vector components) that are later summed up, leading to a total cumulative error

$$Err(\text{SED}) \approx \underbrace{\delta + \delta + \cdots + \delta}_{d \text{ components}} = d \cdot \delta \quad \sim \mathcal{O}(d), \tag{3.9}$$

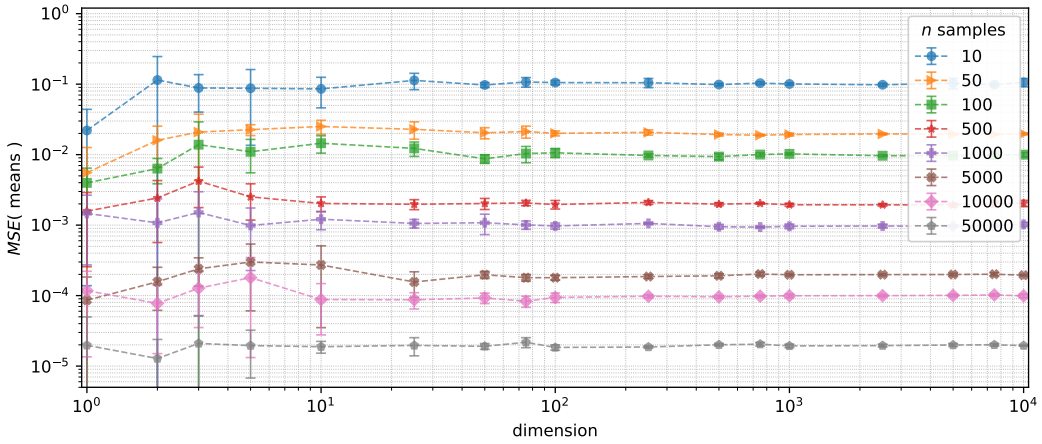
while the Mahalanobis distance formula (matrix times vector) involves d multiplications and additions for each of the d vector components, leading to a total cumulative error

$$Err(\text{MD}) \approx \underbrace{d \cdot \delta + d \cdot \delta + \cdots + d \cdot \delta}_{d \text{ components}} = d^2 \cdot \delta \quad \sim \mathcal{O}\left(d^2\right). \tag{3.10}$$

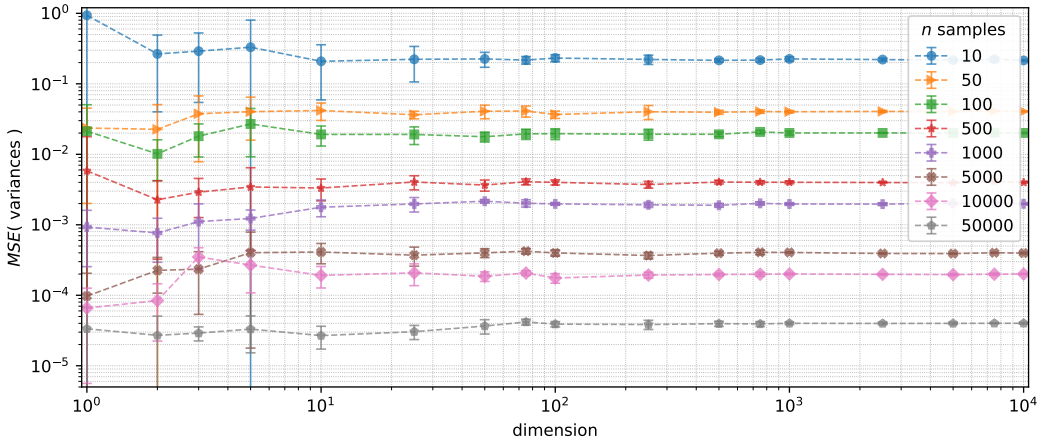
Hence, summarizing, as long as no significant correlation of feature is involved, the SED appears to be preferable than MD, promising more accurate distance calculation in high-dimensional feature spaces.

The obtained results justifies why it is worth to increase the number of training samples n for the estimation of covariance matrix Σ, and hence the utilization of pooled covariance matrix is motivated, i.e., single common covariance matrix assumed for all in-distribution classes (MDP measure, section 2.3.6). However, as it is shown in next chapter, the latter mentioned approach suffers also from the other kind of method error.

(a) Error estimating means



(b) Error estimating variances



(c) Error estimating covariances

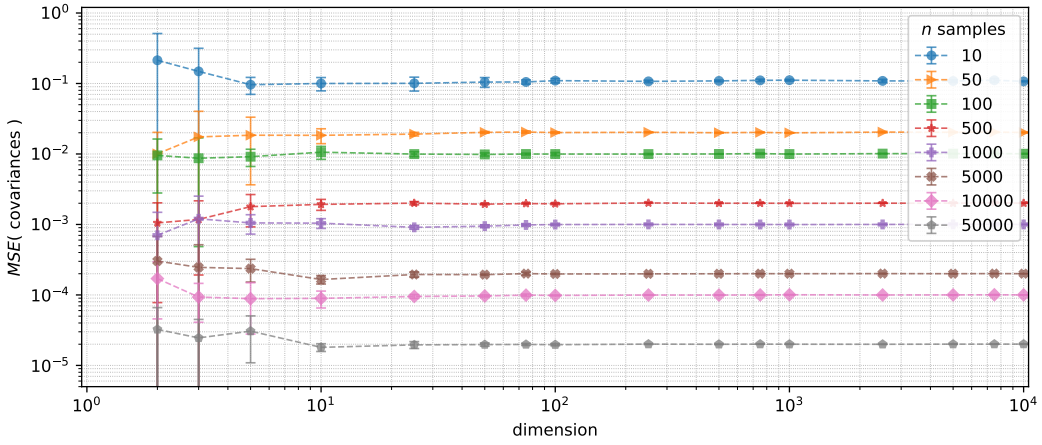
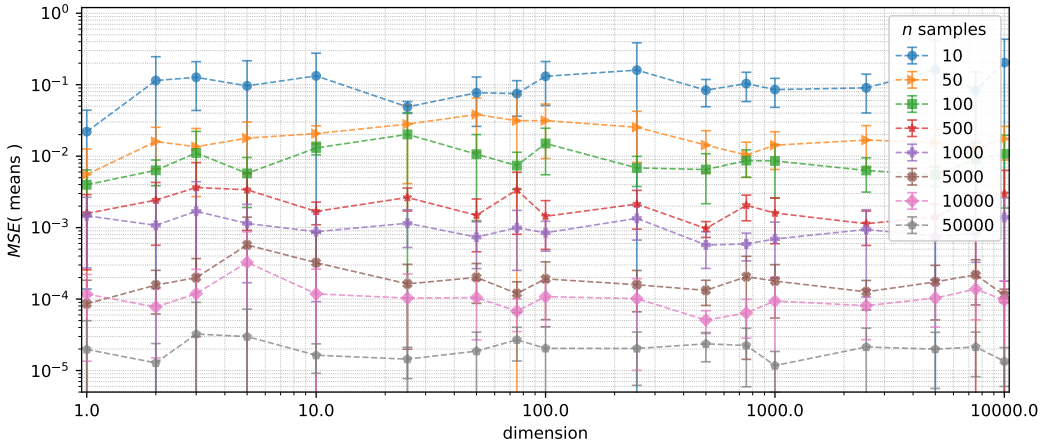
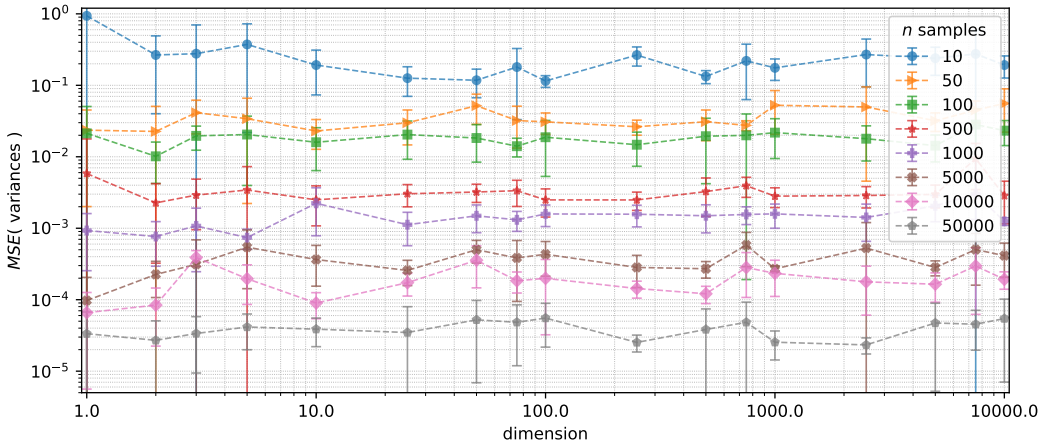


Figure 3.25: The estimation errors of selected cluster properties. The cluster contains n samples of dimension d, generated from $G = MVN$ distribution with small fraction of features $f_{corr} = 0.2$ slightly correlated $g_{corr} = 0.2$. The more n samples provided, the more accurate estimation. The results are aggregated for multiple generator seeds ξ and displayed as averages with error bars (standard deviation).

(a) Error estimating means



(b) Error estimating variances



(c) Error estimating covariances

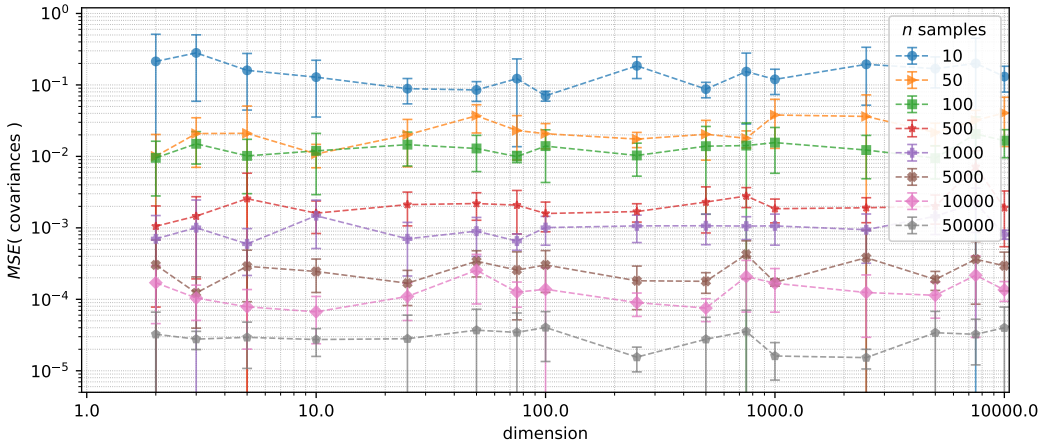


Figure 3.26: The estimation errors of selected cluster properties. The cluster contains n samples of dimension d, generated from $G = MVN$ distribution with great fraction of features $f_{corr} = 0.8$ highly correlated $g_{corr} = 0.8$. Under strong correlation, results are less stable, yet remaining of the same order. The results are aggregated for multiple generator seeds ξ and displayed as averages with error bars (standard deviation).

3.6 Reproducibility of results

All experiments were conducted several times to ensure that the overall patterns and trends were captured, rather than observing a single specific exception – hence the error bars were drawn on plots in this chapter. Additionally, all initialization parameters for the involved pseudo-random number generators (PRNGs) were recorded, so any examined case can be recreated and inspected further in another study, utilizing the tools described in appendix B.

Chapter 4

Performance of OOD detectors in image and text recognition tasks

In this chapter the results of research conducted on image data and text documents are presented. The performance of popular outlier detection methods is shown when applied to feature spaces generated by various Deep Learning (DL) models. It is shown that the effectiveness of OOD detectors are significantly impacted by the characteristics of the feature spaces generated by deep learning representation techniques. Hence, the extended way of evaluating the OOD-generalization properties for selected DL-based data representation methods is proposed. The results are introduced as comparison with traditional measures of efficiency, as being used in current literature. In addition, the differences in properties of the feature vectors obtained from various representation techniques are discussed, resulting in a number of recommendations on the OOD detectors selection.

The described results are base and motivation for further, dedicated publication [15] that was prepared and submitted at the same time as this dissertation chapter.

4.1 Data sources and experiment organization

During the research, the performance of OOD measures in the OOD detection benchmarks was analyzed, involving both image data and text documents as inputs.

For image data, the ImageNet-1K [16] dataset was selected as the in-distribution (ID) data, containing 1‘281‘167 training samples in total, divided into $m = 1000$ classes (i.e., roughly 1300 samples per class). The following datasets were utilized as the source of out-of-distribution (OOD) samples:

- ImageNet-O [30][16],
- iNaturalist [33][17],
- NINCO [4][18],
- OpenImage-O [79][19],
- Places365 [85][20],
- SUN2012 [82][21],
- Textures (Describable Textures Dataset [DTD]) [11][22].

The representation techniques, mentioned in section 2.4, were used to obtain the feature vectors from the images, utilizing models that were pre-trained for the ImageNet-1K dataset (ConvNeXT, EfficientNet, ResNet, ViT) or the Laion2B dataset [36] (CLIP, CoCa). The feature vectors were extracted from the penultimate layer of the pre-trained neural network models.

For each representation, a collection of feature vectors was produced from the ImageNet training data (about 1300 samples per ID class – clusters T_i; i – class identifier), ImageNet validation data (50 samples per class – clusters K_i) and the outliers datasets listed above (clusters U_j; j – dataset identifier). Then, for each ImageNet class i:

- The OOD detector with selected outlierness measure OF was fitted to the feature vectors corresponding to the class training data T_i.

[16] https://github.com/hendrycks/natural-adv-examples
[17] https://github.com/visipedia/inat_comp
[18] https://github.com/j-cb/NINCO
[19] https://github.com/haoqiwang/vim
[20] http://places2.csail.mit.edu
[21] https://groups.csail.mit.edu/vision/SUN/hierarchy.html
[22] https://www.robots.ox.ac.uk/~vgg/data/dtd/

- Using the fitted OOD detector, the outlierness scores were calculated for each of the obtained feature vectors from:
  - the class training data T_i,
  - the class validation samples K_i,
  - all data from the selected outliers/OOD dataset U_j.
- The AUROC values between the scores obtained for clusters K_i and U_j were calculated (determining the separability between ID and OOD data).
- The sensitivity and specificity measures were computed, as proposed in section 2.2.4 – to determine the performance of ID samples recognition (sensitivity – using K_i data) and OOD detection (specificity – using U_j data) when the detection threshold t is calibrated so that 95% of ID training samples (T_i data) are recognized correctly as in-distribution.

For text documents, there are no standard datasets available that are widely used for outlier/OOD detection benchmarks, hence in experiment both the ID and OOD examples were selected by class-wise division of utilized dataset. The study on text data was conducted on two kinds of documents: long (e-mails) and short (sentences).

- For long documents, the 20newsgroups dataset [41][23] was used[24], containing around 18000 documents from 20 labeled topic categories. The dataset was arbitrary divided into ID samples (17 categories) and OOD data (3 categories).
- For short documents, the banking77 dataset [9][25] was used, containing about 13000 sentences (customer service queries) labeled with 77 classes (customer intents). The dataset was randomly divided into ID samples (62 classes) and OOD examples (15 classes).

The feature vectors from the text documents were produced using the pre-trained BERT (2 variants: base and tiny) and fastText models; in case of Doc2Vec and TF-IDF the models were built based on the training data. The calculation of AUROC scores and classification with respect to the training samples were performed analogously how the image data were analyzed.

[23] http://qwone.com/~jason/20Newsgroups/

[24] https://scikit-learn.org/stable/datasets/real_world.html#the-20-newsgroups-text-dataset

[25] https://github.com/PolyAI-LDN/task-specific-datasets

In the conducted research the OOD detectors involved following OF measures: kNN, LOF, MD, MDP and SED. The angle-based OOD detectors, described in section 2.3, are not commonly used in large-scale image recognition benchmarks (such as ImageNet) due to their high computational cost. Three of the studied measures are similar, based on the (co)variances estimation:

- MD – involves the estimation of full covariance matrix to capture the correlations in data, calculated per each known ID class, hence each matrix is produced from relatively small number of samples, which results in instability and higher estimation errors (section 2.3.6).
- MDP – modifies the MD by utilization of the pooled covariance matrix, i.e., one common covariance matrix is calculated for data from all m known ID classes (section 2.3.6).
- SED – assumes no correlations in data, which corresponds to the diagonal covariance matrix, i.e., only the axis-wise variances are calculated, improving the stability of the distance calculation (section 2.3.7).

The MDP variant of Mahalanobis distance is a standard widely used and recommended in the OOD detection literature [43][21][65][20] – as an approach to increase the number of samples involved in covariance matrix calculation, effectively reducing the estimation error (section 3.5.2). However, as shown in later the sections, this approach results in a new kind of error, due to the fact that ID classes are characterized by various correlation degrees, effectively making it one of the worst solutions of all studied.

4.2 Performance of OOD detectors for different representation spaces

The table 4.1 contains the average AUROC scores, calculated for 5 analyzed outlierness measures OF (section 2.3), specified by the representation model used to obtain the feature vectors, as well as the selected outlier data. AUROC is the commonly used measure in literature to conveniently summarize the overall performance of the outlier detection techniques, such as visible in benchmark published by Yang et al. [83]. A general dependency between the chosen representation and favorable outlierness measure for that representation can be observed.

The most important observation is that the performance of OOD detector is related to the utilized representation model and independent of the analyzed outlier data collection. Some representations favor specific OOD detectors, while other OOD detectors perform worse on given representations – in some cases, notably for EfficientNet and ResNet, the differences in OOD detectors performance are significant.

The following general conclusions can be made:

- For ResNet: the best measure appears to be SED, while the worst is MDP.
- For ViT: LOF and MD offer best performance, SED and MDP turn out the worst.
- For EfficientNet: kNN performs best, LOF and MDP have the worst results.
- For ConvNeXT: SED appears best, along wit kNN, while LOF and MDP – worse.
- For CLIP and CoCa: kNN outperforms other measures, SED is very close to top, while MDP obtains the worst results in all cases.

Overall, the CLIP and CoCa representations offer the highest AUROC scores, making those the best in terms of OOD-generalization, no matter what kind of outliers dataset was used. It is also worth to notice that MDP, although commonly recommended in literature, performed bad in general – it achieved the worst results in most cases and it is never better than MD or SED (except for ViT).

However, it turns out that such results presentation effectively hides some facts that are especially important from the safety-critical applications. Hence, the more detailed analysis is useful, presented in next section.

outlier data	measure	CLIP	CoCa	ConvNeXT	EfficientNet	ResNet	ViT
ImageNet-O	kNN	**0.998**	**0.996**	**0.977**	**0.966**	0.766	0.915
	LOF	0.986	0.979	0.971	0.873	0.770	**0.957**
	MD	0.993	0.982	—	0.946	—	0.952
	MDP	*0.955*	*0.935*	*0.943*	*0.761*	*0.608*	0.909
	SED	**0.998**	0.993	**0.977**	0.932	**0.877**	*0.901*
iNaturalist	kNN	0.998	**0.996**	**0.953**	**0.984**	0.777	0.931
	LOF	0.986	0.974	*0.925*	*0.756*	0.727	0.969
	MD	0.990	0.981	—	0.976	—	**0.977**
	MDP	*0.885*	*0.897*	0.926	0.844	*0.487*	0.948
	SED	**0.999**	0.995	0.948	0.917	**0.903**	*0.922*
NINCO	kNN	**0.998**	**0.996**	0.963	**0.975**	0.738	0.938
	LOF	0.987	0.976	0.937	0.814	0.711	0.971
	MD	0.992	0.981	—	0.957	—	**0.974**
	MDP	*0.931*	*0.916*	*0.936*	*0.758*	*0.452*	0.943
	SED	**0.998**	0.994	**0.965**	0.917	**0.874**	*0.928*
OpenImage-O	kNN	**0.999**	**0.998**	0.959	**0.974**	0.780	0.950
	LOF	0.981	0.978	*0.895*	*0.777*	0.742	**0.975**
	MD	0.995	0.986	—	0.956	—	0.974
	MDP	*0.963*	*0.949*	0.933	0.782	*0.518*	0.942
	SED	**0.999**	0.996	**0.966**	0.883	**0.886**	*0.941*
Places365	kNN	**0.999**	**0.998**	0.964	**0.977**	0.783	0.939
	LOF	0.982	0.979	*0.915*	*0.755*	0.740	**0.969**
	MD	0.994	0.984	—	0.960	—	0.968
	MDP	*0.961*	*0.941*	0.933	0.808	*0.505*	0.928
	SED	**0.999**	0.994	**0.969**	0.897	**0.887**	*0.927*
SUN2012	kNN	**0.999**	**0.997**	0.965	**0.974**	0.765	0.931
	LOF	0.984	0.979	0.930	0.761	0.710	**0.965**
	MD	0.995	0.982	—	0.950	—	0.964
	MDP	*0.967*	*0.929*	*0.929*	*0.760*	*0.429*	*0.916*
	SED	0.998	0.990	**0.969**	0.893	**0.875**	0.919
Textures	kNN	**0.999**	**0.995**	0.982	**0.983**	0.840	0.943
	LOF	0.989	0.977	*0.948*	*0.807*	0.829	0.971
	MD	0.995	0.976	—	0.974	—	**0.975**
	MDP	*0.972*	*0.923*	0.964	0.890	*0.729*	0.948
	SED	0.998	0.989	**0.988**	0.941	**0.922**	*0.933*

Table 4.1: Average AUROC scores observed for various outlierness measures OF, calculated between the in-distribution data (ImageNet1K) and 7 other datasets (outliers), utilizing different representation generators for feature vectors. In each column the result for the best OOD detector is marked with **bold**, and the worst observed result – with *italic*.

4.3 Analysis of the per class OOD-generalization

In the previous section, performance on OOD detection methods in different representation spaces was analyzed using the overall AUROC measure. This is the standard metric in current literature for evaluating the performance of OOD detectors in OOD detection benchmarks.

This section proposes an essential extension of OOD evaluation to go beyond the overall AUROC measure in favor of per-class analysis – i.e., presenting AUROC scores calculated per individual ID classes. This evaluation method allows the identification of the weak or worst-case ID classes, i.e., classes with low AUROC scores. This finding indicates a severe security gap in the deep model due to these classes, which realize low OOD generalization, i.e., are easily confused with OOD samples. In this way, the practitioners who implement AI models get additional insight into the safety risks of models deployed in safety-critical applications.

Because the research was performed per known in-distribution class, a detailed illustration of the reached AUROC scores per representation and utilized outlierness measure can be created. Figure 4.1 presents the observed AUROC scores using bounding boxes for better understanding of the performance and measures/representation relations. The boxplots covers the median and upper/lower quarterly (50% of observations), while the whiskers indicate the range of 95% of results; the average values are marked with white dots for the reference. The key observations from this figure are:

- The performance of the outlierness measures is related to the utilized representation model. The detector that performed well for some representation, may be notably worse when used with feature vectors produced by other model – e.g., SED outperforms other measures for ResNet, while for ViT it turns to be the worst.
- In each case there exists significant number of classes that fall behind in terms of separability from outliers. Such classes can introduce security gaps in real-world applications of models and should be examined in safety-critical implementations.
- The models pre-trained on a large collection of image+text pairs (CLIP, CoCa) clearly outperformed other models in the task of separability. The ConvNeXT and ViT representations can also provide high-quality results, depending on the utilized outlierness measure. The ResNet turned out the worst in this task.

The results in figure 4.1 aggregate data for all classes from table 4.1, while figure 4.2 contains the selected results per outlier data. No major differences can be observed between the analyzed dataset, indicating that the observed separability depends more

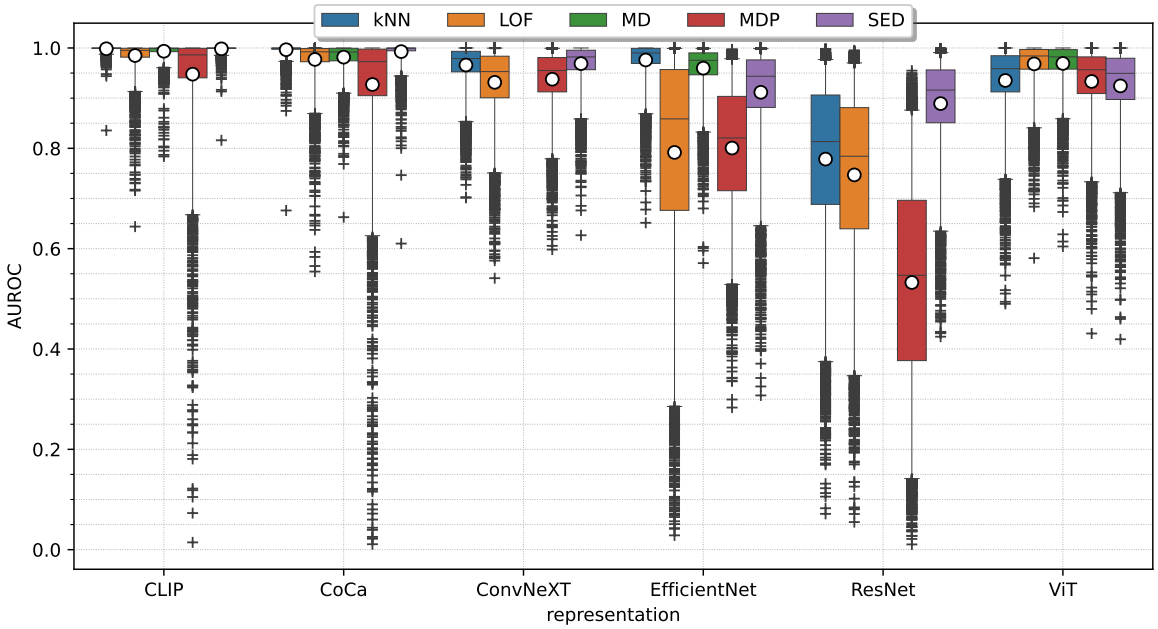


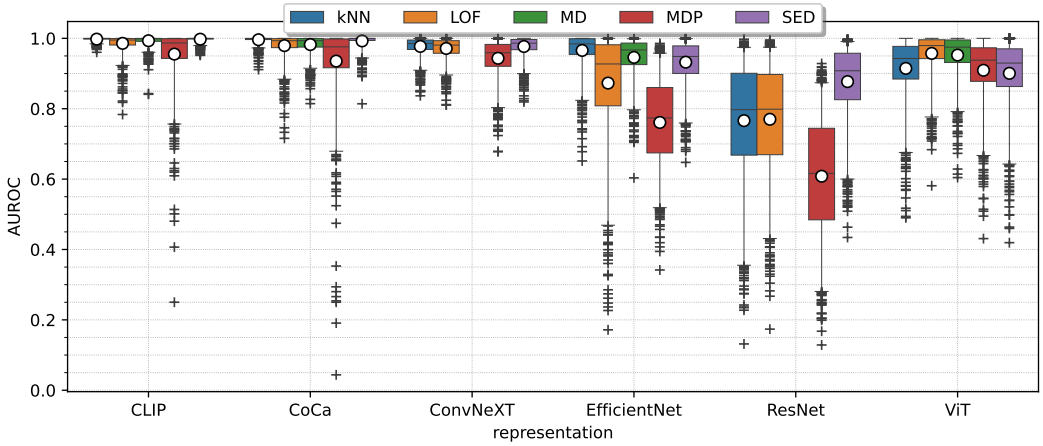
Figure 4.1: Observed separability between in-distribution data (ImageNet samples) and the out-of-distribution data from 7 datasets: ImageNet-O, iNaturalist, NINCO, OpenImage-O, Places365, SUN2012 and Textures. White dots mark the average scores, while the whiskers indicate the range of 95% AUROC values calculated for a given representation and outlierness measure. For some classes (marked with crosses) the calculated AUROC value is very low, making them indistinguishable from outliers.

on the utilized feature vectors representation and involved outlierness measure, rather than the examples that were examined. This is test This is test This is test This is test This is test This is test

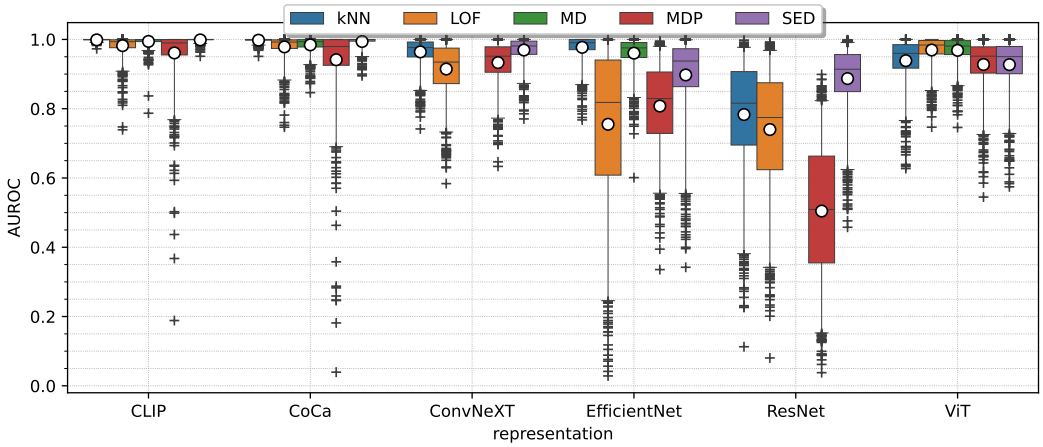
The results obtained for text documents are presented in figure 4.3a and 4.3b. Similar conclusions can be drawn from the research – kNN perform best when used with BERT models, but falls out when used with fastText, where LOF offers the top-result, at the same time being the worst to use with BERT. In addition, it appears that the kind of utilized ID corpus plays the significant role as well – the separability (AUROC scores) was much better in case of short documents (sentences) rather than for longer documents (e-mails).

The lack of results for Mahalanobis distance (MD) in case of ConvNeXT and ResNet representation is a result of problems with covariance matrix estimation. Both representations utilize feature vectors of dimensions (section 4.5, table 4.2) greater than the number of training samples available in the ImageNet dataset. The alternative suggested in literature in such cases is to utilize the Mahalanobis distance with pooled covariance matrix (MDP), however this approach turned out to be the worst possible overall in all analyzed cases.

(a) Separability: ImageNet (ID) vs. ImageNet-O (OOD)



(b) Separability: ImageNet (ID) vs. Places365 (OOD)



(c) Separability: ImageNet (ID) vs. SUN2012 (OOD)

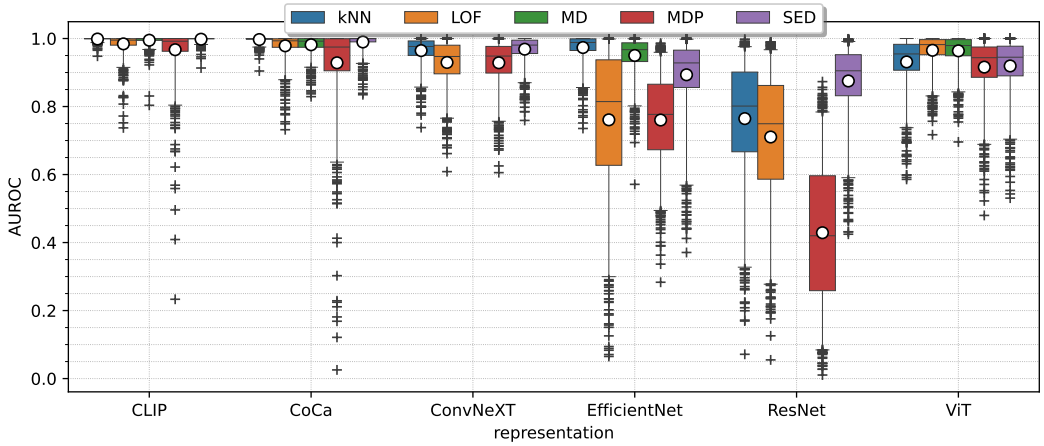
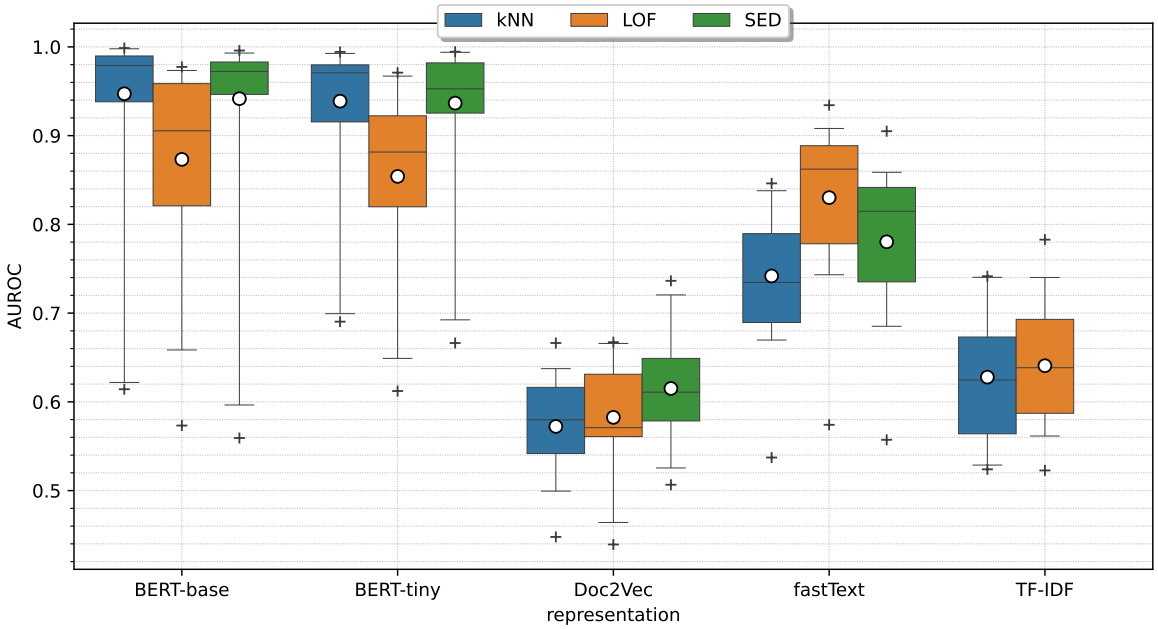


Figure 4.2: Detailed comparison of obtained AUROC scores per selected OOD data. No major differences can be observed between the analyzed datasets, the utilized representation plays major role along with the chosen outlierness measure.

(a) Separability: 20newsgroups (17 ID classes vs. 3 OOD classes)



(b) Separability: banking77 (62 ID classes vs. 15 OOD classes)

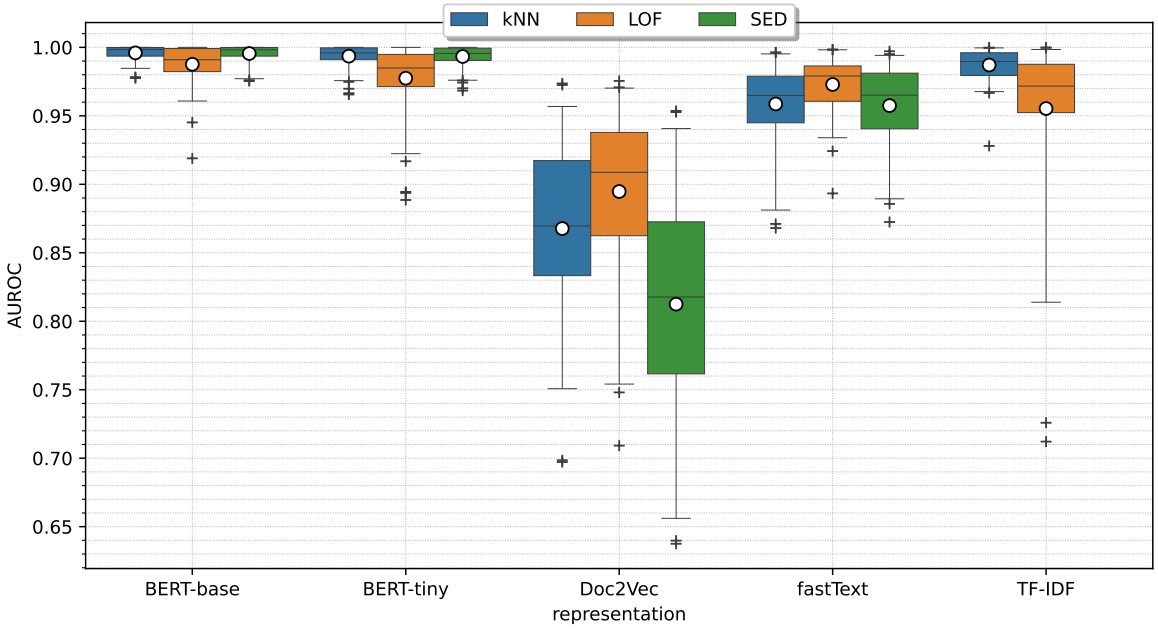


Figure 4.3: Calculated AUROC scores, obtained in the separation task between the text data. In case of short documents (banking77) the separation task turns out to be much easier for all analyzed representations (all scores AUROC $\gtrsim 0.7$). The Doc2Vec and TF-IDF representations for 20newsgroups do not provide vectors suitable for performing the out-of-distribution detection.

4.4 Performance of OOD detectors calibrated on training ID data

Figure 4.4 presents the analysis of recognition capabilities for outlierness measures and utilized representation models. The classification of testing samples was performed with respect to the training data – calibrated so that the 95% of training samples are properly recognized as in-distribution data. Just like in case of AUROC scores, the obtained scores for each measure differ between representations and no single universal recommendation can be given.

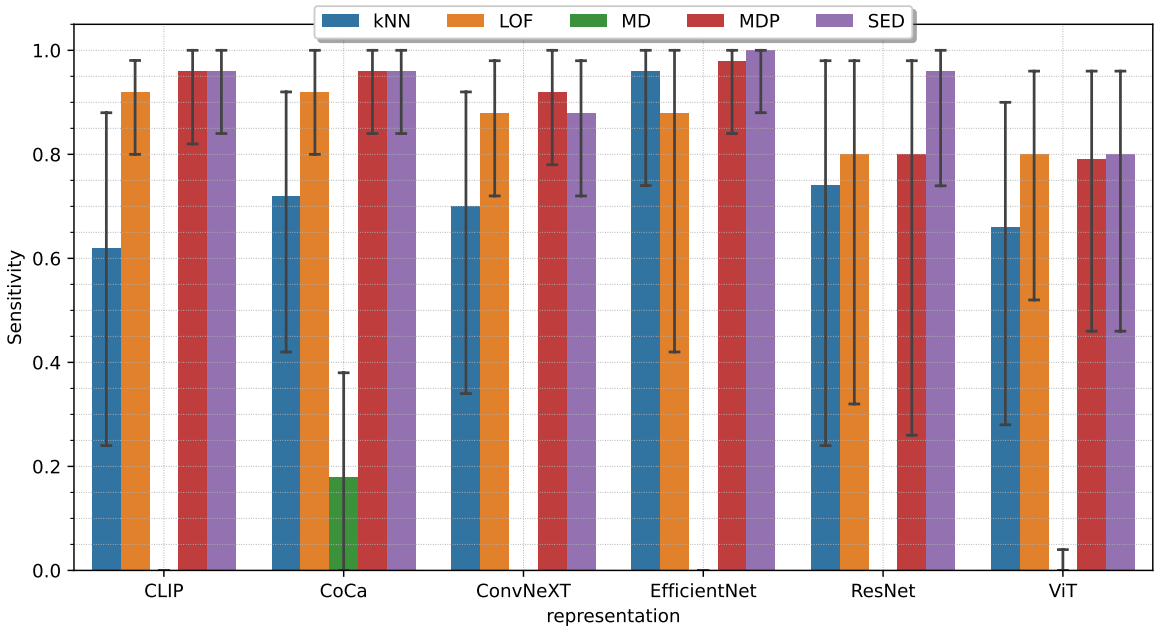
The results show similar observation as made with the numerical simulations (chapter 3) for MD – in considered experiment layout (ImageNet as training data), the available number of training samples n was too low for the dimension d of feature space, so the measure's sensitivity was dramatically low. Under the given criteria, i.e., recognition with respect to the training samples, MD spuriously recognizes all in-distribution testing samples as outliers. However, as per figure 4.1, the MD offers satisfying separability between ID and OOD data (i.e., reaching one of the top AUROC scores), at least in cases where it was possible to utilize (except ConvNeXT and ResNet). This suggest a potential for MD being a fine OOD detector, although a practical utilization in such conditions would requite the threshold calibration using validation data – which is still at least questionable recommendation in the context of robustness and safety-critical applications.

An alternative commonly proposed in literature for such cases is the utilization of MD variant with pooled covariance matrix – MDP. This variant performs surprisingly well in terms of in-distribution samples recognition (sensitivity), while also maintaining satisfying specificity in most of the cases (except EfficientNet and ResNet). Yet, the best outlierness method overall in this task appears to be SED, reaching top scores for both sensitivity and specificity for every representation except for EfficientNet, where the best measure turned out to be kNN.

The result of classification performed on text documents are presented in figure 4.5 for completeness. The best metric overall appears to be LOF here, competing closely with SED and kNN for BERT representation. Notably the observed scores are higher for classification task on the short text documents (banking77).

Note the accuracy metric is not presented in this chapter, as due to experiment organization (50 ID samples vs thousands of OOD examples) the results would be heavily biased towards the specificity value anyway. Hence, only the sensitivity and specificity metrics are used for comparison.

(a) Correctly recognized in-distribution samples (ImageNet)



(b) Correctly recognized out-of-distribution samples (outliers)

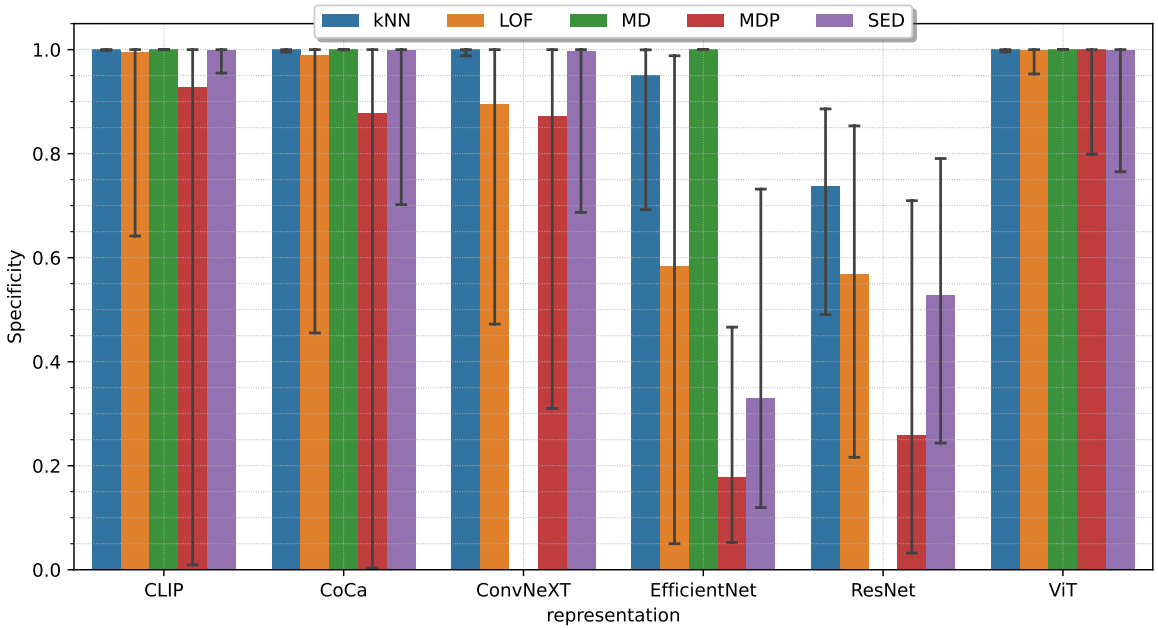
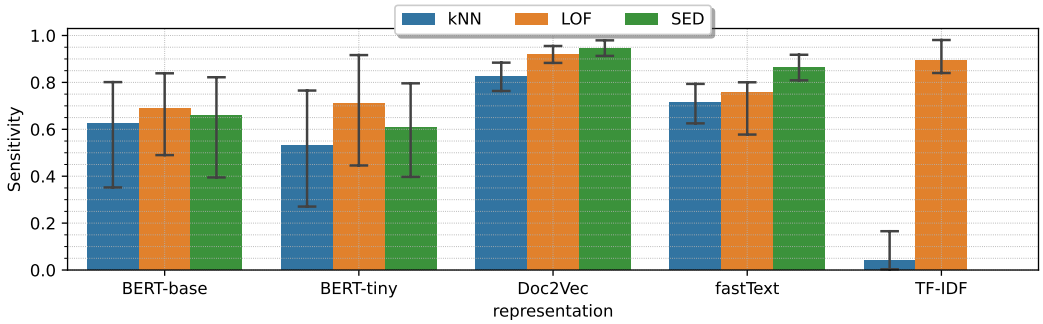
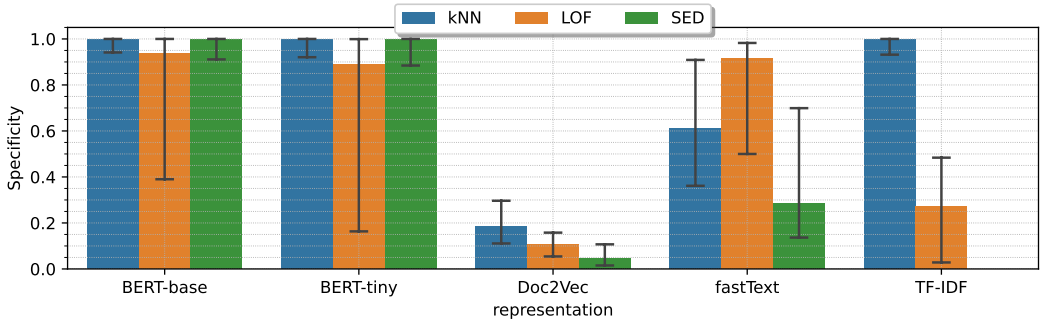


Figure 4.4: Results of the image data classification with respect to the training samples. Rejection threshold set so that 95% of the training data would be correctly recognized as in-distribution. Outliers consist of samples from 7 datasets: ImageNet-O, iNaturalist, NINCO, OpenImage-O, Places365, SUN2012 and Textures.

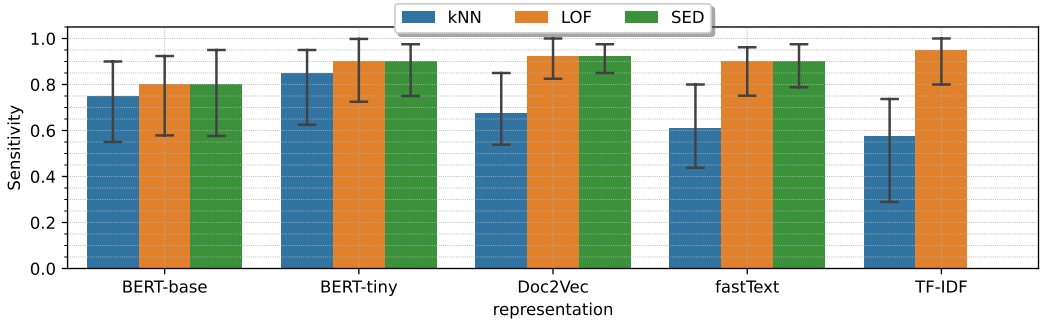
(a) Correctly recognized in-distribution samples (20newsgroups)



(b) Correctly recognized out-of-distribution samples (20newsgroups outliers)



(c) Correctly recognized in-distribution samples (banking77)



(d) Correctly recognized out-of-distribution samples (banking77 outliers)

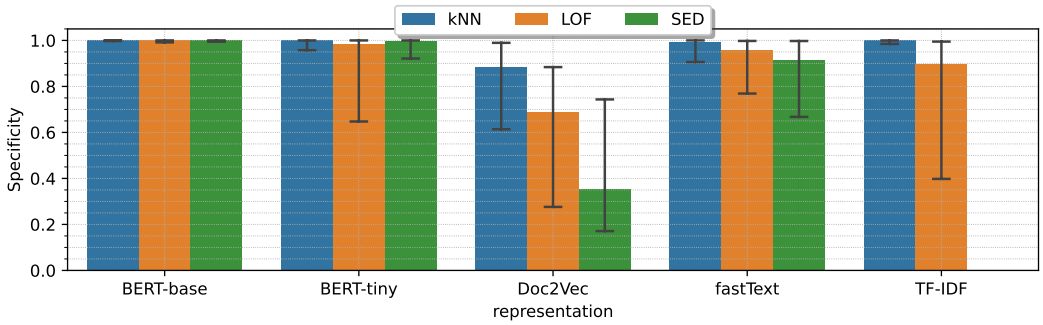


Figure 4.5: The classification of text documents with respect to the training datasets. Rejection threshold set at $TPR = 95\%$ of the scores obtained for training data.

4.5 Characteristics of feature vectors

This section contains the analysis and discussion on the properties of the data representations generated by various Deep Learning models – and how these properties affect (or not) the performance of OOD detection methods. The results correspond to observations made in previous sections (e.g., table 4.1, figures 4.1, 4.3, 4.4 and 4.5).

The table 4.2 summarizes the dimensionality of feature vectors for image data – coming from the utilized representations models (CLIP, CoCa, ConvNeXT, EfficientNet, ResNet, ViT). The table 4.3 covers analogous information regarding the feature vectors obtained for text documents from models (BERT, Doc2Vec, fastText, TF-IDF).

Figures 4.6, 4.7, 4.8 illustrate the various discovered properties of the feature vectors, per in-distribution data class and utilized representation model, obtained for the image data (ImageNet), long (20newsgroups) and short (banking77) text documents, respectively. The considered values are as follows:

- variance – calculated as an average value over all features in a given class,
- correlation – an average of the absolute values calculated within the top triangle of the covariance matrix,
- kurtosis – computed as an average over all features for each class; value 3.0 corresponds to normal-like distribution, big values indicate long tails, low values are related to highly-concentrated data.
- skewness – also calculated as an average value over all features; values in range $(-0.5, 0.5)$ mean that distribution is approximately symmetric, for values $(0.5, 1.0)$ the distribution is slightly skewed and for values $(1.0; \infty)$ it is considered a highly skewed distribution.
- test of normality – average p-value obtained for each feature in class, using the D'Agostino and Pearson's test[26] with assumption of the Gaussian distribution as the null-hypothesis.
- number of feature-wise outliers – the average number of observations that fall outside the confidence interval (range ± 1.5 IQR threshold) based on the values of each feature in the training clusters vectors.

Important observation for all the analyzed cases is that, in general, there are no strongly correlated features – all medians of absolute correlations are below 0.25. However the observed correlations vary per class, especially significantly for some

[26] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.normaltest.html

representation	CLIP	CoCa	ConvNeXT	EfficientNet	MobileNet	ResNet	ViT
dimension d	1024	769	1536	1280	1280	2048	1024

Table 4.2: Dimensionality of the feature vectors used for image representations.

representation	BERT-base	BERT-tiny	Doc2Vec	fastText	TF-IDF (20newsgroups)	TF-IDF (banking77)
dimension d	768	128	300	100	5000	2095

Table 4.3: Dimensionality of the feature vectors used for text representations.

representations (EfficientNet and BERT), and this raises the question of justification for utilization of the pooled covariance matrix. As classes present various correlations, the single covariance matrix incorrectly reflects data from some of these classes, i.e., it results in an important method error, visible as low performance of MDP measure.

The features produced by ResNet are especially weakly correlated, which corresponds with high efficiency of SED measure with this representation in conducted experiments. Contrary, EfficientNet is characterized by higher correlations, hence the MD performs better than SED in that case (i.e., ignoring the correlations leads to worse performance).

Similarly, there are no significant differences in average variances of features observed. Although the BERT, CoCa and fastText produced feature vectors of especially low average variance. Most representations produce features that appear symmetric (BERT, CLIP, CoCa, ConvNeXT, Doc2Vec, fastText and ViT), with exception of EfficientNet and ResNet, where the valued are highly skewed – and additionally, in case of ResNet, widely distributed with long tails.

For TF-IDF the SED measure was impossible to utilize due to multiple empty features in obtained vectors, i.e., zero-variance calculated and therefore division by zero encountered in formula 2.29 (section 2.3.7). Also, because of the TF-IDF feature vectors sizes, the estimation of properties such as correlations was time-consuming and hence it is omitted in the figures.

It must be noticed that the characteristics obtained for short and long text documents differ significantly in some cases, e.g., in the normality tests (figures 4.7d and 4.8d). This corresponds with inconsistencies observed fpr results of AUROC, sensitivity and specificity scores between 20newsgroups and banking77, presented in previous sections. Hence, a more detailed study of the relation between the characteristics and the ID data shall be conducted in the future.

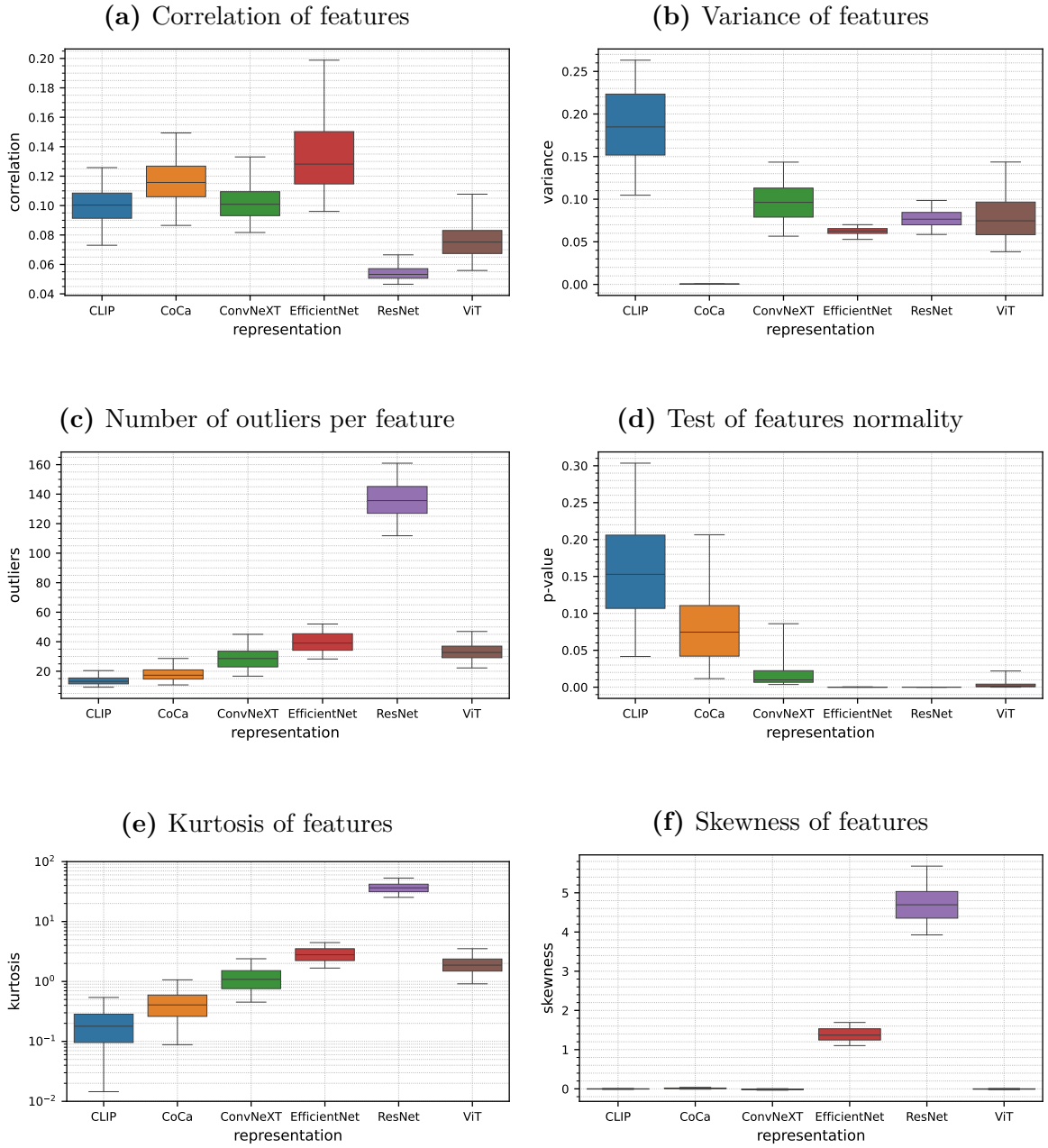


Figure 4.6: Properties of feature vectors corresponding to ImageNet classes obtained for 6 different representations algorithms (CLIP, CoCa, ConvNeXT, EfficientNet, ResNet and ViT). The same image data can be expressed as a completely different feature vector, depending on the chosen representation.

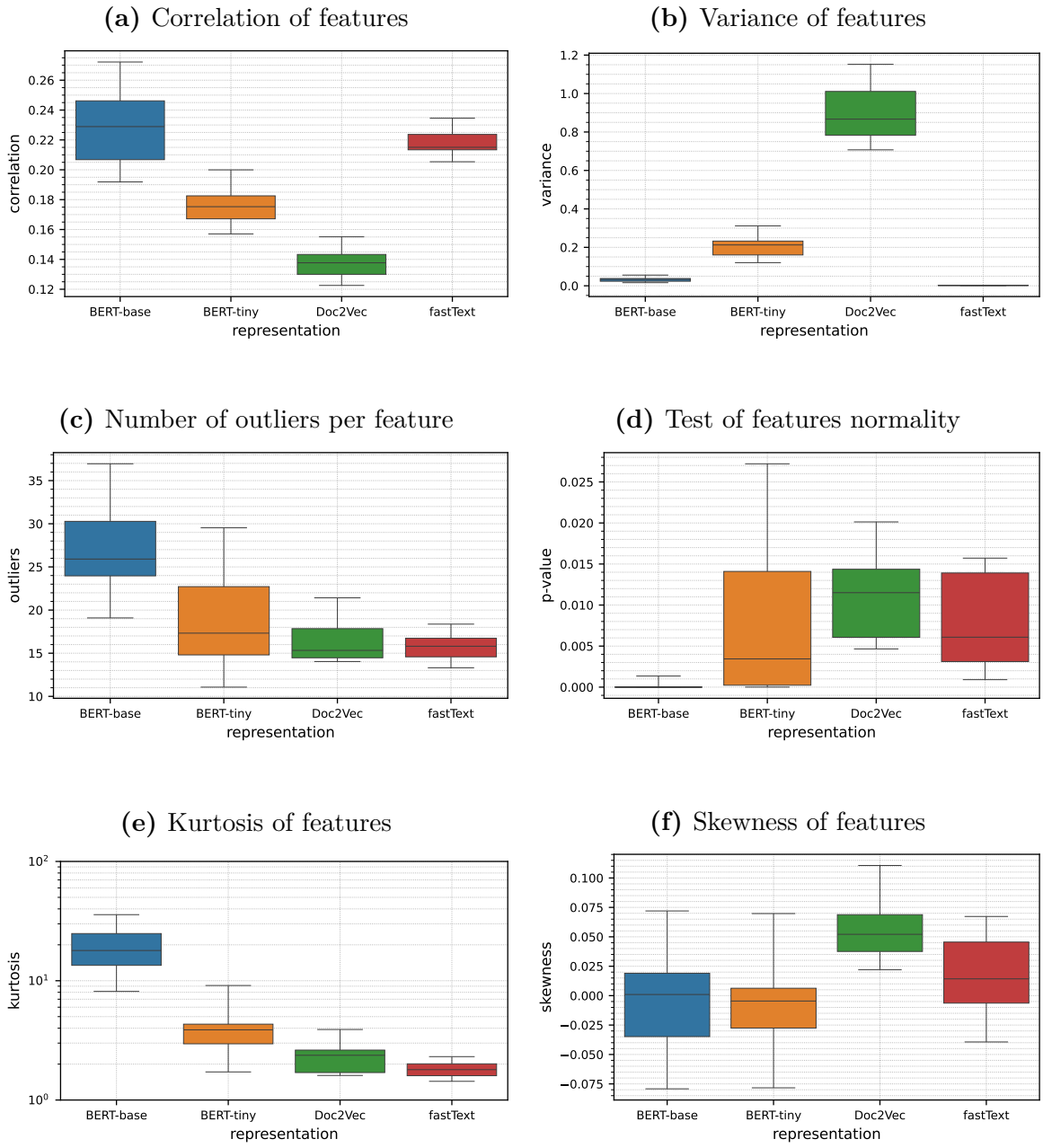


Figure 4.7: Properties of feature vectors corresponding to 20newsgroups classes obtained for different representations algorithms (BERT, Doc2Vec and fastText). Analysis for TF-IDF was omitted due to the size of feature vectors.

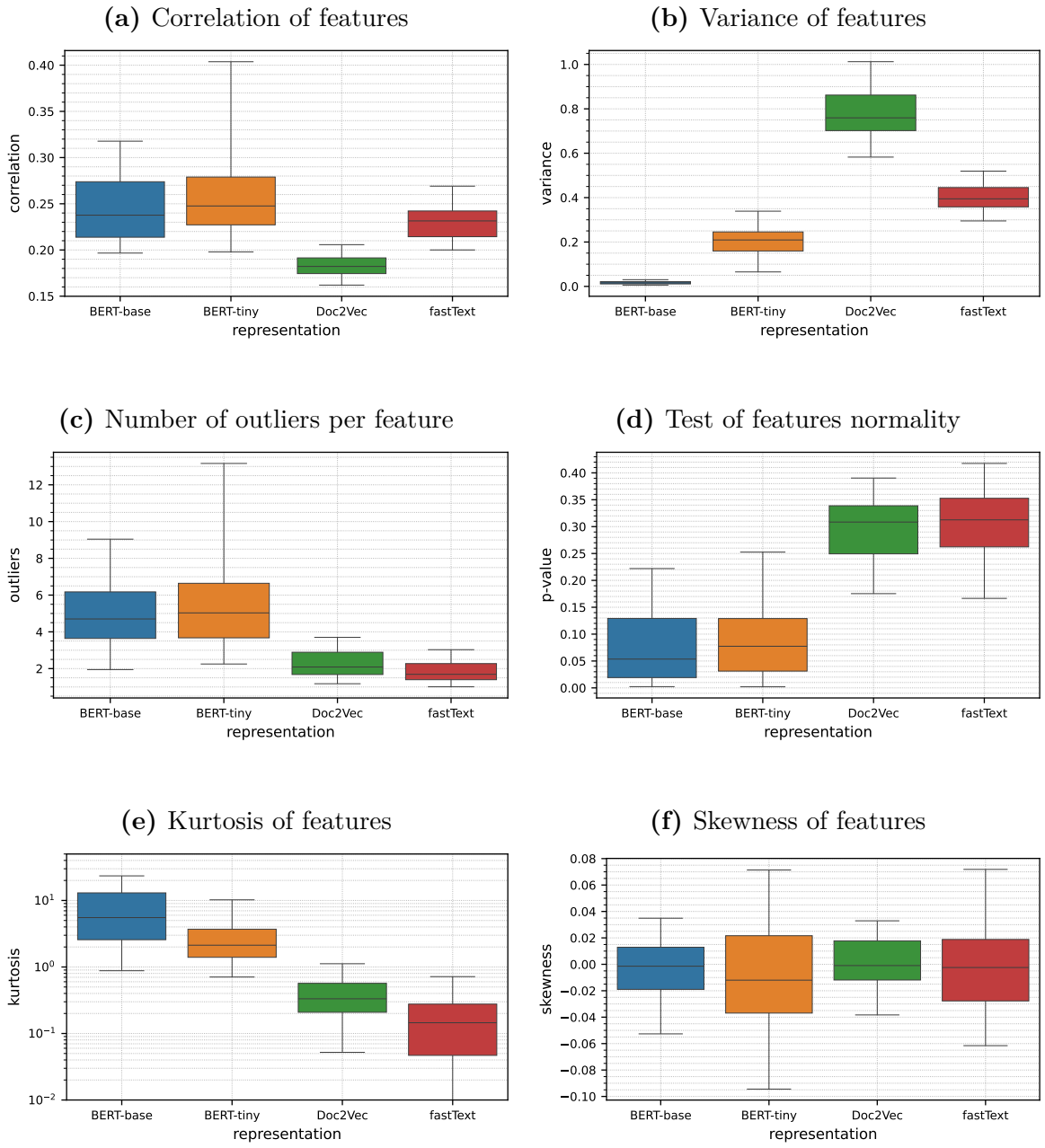


Figure 4.8: Properties of feature vectors corresponding to banking77 classes obtained for different representations algorithms (BERT, Doc2Vec and fastText). Analysis for TF-IDF was omitted due to the size of feature vectors.

Chapter 5

Summary

In this work the problem of open-set classification in high-dimensional feature spaces was explored. It is a key task to be solved to ensure the security of implementations and the reliability of machine learning models – in particular: deep learning models for image and text recognition; allowing the systems that utilize such models to react predictably to new data and unexpected situations. Although the problem remains open, the conducted research contributes to the field, introducing new insights of the selected *post-hoc* methods behaviors and properties, as well as providing recommendations of outlierness measures and high-dimensional representation techniques for applications involving image and text data.

A comprehensive study on the performance of selected *post-hoc* methods for outliers detection was conducted. Primarily, it was shown that the performance of OOD detectors is dependent on the properties of the representation – i.e., of the feature space in which the recognition of outliers is performed. This problem is not sufficiently noticed in the existing literature – the rankings of methods in benchmarks are presented, comparing results for various representations or showing results for a fixed selected representation (usually ResNet) without emphasizing it, making the formulated general conclusions turning out to be not useful when new representations or out-of-distribution detection methods are analyzed.

One of the most interesting observation is related to the phenomenon that the measures can consider training and testing data coming from the same distribution (ID) as distant from each other, i.e., testing data may be considered outliers with respect to the available training samples. For ED, SED and IRWD measures it is only observed for highly under-represented training cluster for a given dimension, i.e., $n \ll d$, and quickly vanishes as number of training samples grows. For MD measure this phenomenon is observed unless a significantly great number of training samples is provided, $n \gg d$,

which in high-dimensional feature spaces becomes troublesome. In case of kNN, the phenomenon is also observed and it is related to the selected value of k – for low value k the train-test distance is greater than for higher values of k. Detailed research of this effect for kNN shall be conducted in the future. For ABOF and LOF measures this phenomenon was not observed even under extreme conditions.

Despite the observed train-test data distancing, all analyzed outlierness measures can still reliably distinct the in-distribution (ID) data from sufficiently distant out-of-distribution (OOD, outliers), which is proven by the calculated AUROC scores. However, this means that in some cases, notably for MD and kNN, the threshold calibration for the open-set classification task requires involvement of the additional validation in-distribution data to achieve correct results, although reliability of such approach may be questionable.

For all analyzed measures, the more distant the outliers were from the in-distribution samples, the higher AUROC scores were observed. Contrary, higher dimension d of feature vectors made the separability between in-distribution and out-of-distributon data more challenging. The number of training samples n did not affect much the methods performance for a given fixed d and h, except for the extreme conditions (e.g. $n < 10^2$ for $d = 750$) or MD measure, where there is a computational requirement $n \geq d$ (condition for inverting the covariance matrix used in distance formula).

The results of numerical study were compared with the open-set classification task conduced on the real-world data, utilizing feature vectors produced by multiple representation algorithms – CLIP, CoCa, ConvNeXT, EfficientNet, ResNet and Vit for image data; BERT, Doc2Vec, fastText and TF-IDF for text documents. Future work in this field shall conduct more comprehensive study, involving the research focused on other domains than image and text recognition – for example AI in medicine, such as the bacteria identification based on genomic sequences [56][21], where the OOD detection is the essential element in real-world applications.

It was shown that for all analyzed cases there exist a notable number of classes that contain samples much more difficult to distinguish from outliers. Such classes pose an important security gap in the deployed machine learning system. Hence, the per-class analysis of ID-OOD separability is proposed as a recommended approach any safety-critical applications. Further work in this topic should involve a detailed focus on those classes – the curious question remains if those are the same classes for all outlierness measures and representation algorithms, and notably why such classes are characterized by poor performance in the outliers detection task.

5.1 Recommendations for OOD detection with Deep Learning models

Based on the conducted study, the following recommendations regarding the usage of OOD detection methods can be formulated, assuming the task in image and text recognition domain using Deep Learning representation methods.

1. The OOD detector based on the Mahalanobis distance with a single, common covariance matrix for all classes of the known ID set (i.e., pooled variant, MDP) should be avoided and not used – it is always less effective in outlier detection task than the original variant (MD), which involves a separate covariance matrix calculated for each ID class, or the variant with diagonal covariance matrix (SED), which assumes no correlations between features. The selection between MD and SED can be made by analysis of the representation characteristics – for model with no or low correlations of features, such as ResNet, the SED is preferred.

2. The operating point, i.e., OOD detection threshold (formula 2.3), of detectors based on popular kNN and MD measures, cannot be calibrated on the condition that a given fraction of training data (e.g., 95%) will be correctly recognized. For such measures, the procedure results in a low sensitivity of ID recognition by the detector. An additional, independent set of validation data must be used to calibrate detectors based on such measures. However, other OOD detectors, e.g., utilizing LOF and SED, can be successfully calibrated on just the training in-distribution (ID) data.

3. The selection of OOD detector in real-world implementations of the Machine Learning systems should be made based on the analysis of the representations characteristics generated by the Deep Learning models. For example, for the ResNet model the preferred OOD detector involves SED measure, while for the ViT model both LOF and MD performs best. The OOD detector rankings and recommendations made for a fixed representation (Deep Learning model) do not transfer to other models.

4. The data representation models generated by various Deep Learning techniques (e.g., CNN, ViT, CLIP) differ significantly in therms of OOD-generalization, i.e., the susceptibility to errors of incorrect identifications of OOD observations. Hence, in any safety-critical task, it is recommended to utilize either CLIP, CoCa or ConvNeXT models, as offering the best results in general. Models such as ResNet, EfficientNet and ViT bear a greater risk of OOD data incorrect recognition.

5. The implementation of the AI system in the real-world tasks, especially for safety-critical ones, should be preceded with an analysis of OOD-generalization per every known class. This will allow to identify the security gaps in the system by capturing the classes with low OOD-generalization, i.e., classes that are easier to confuse with out-of-distribution data, i.e., are more prone to incorrect recognition.

Relying on these above recommendations will improve the quality of OOD observations recognition, therefore resulting in improved trustworthiness of Machine Learning systems in real-world and safety-critical applications.

Appendix A

Glossary

ABOF – **A**ngle-**B**ased **O**utlier **F**actor
A measure to quantity the similarity of the data.
See: section 2.3.1

BERT – **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
A transformer model designed for NLP tasks.
See: section 2.4.8

BoW – **B**ag **o**f **W**ords
A way to represent text documents as feature vectors.

CLIP – **C**ontrastive **L**anguage-**I**mage **P**re-training
A method for image classification with transformer-based architecture.
See: section 2.4.1

CoCa – **Co**ntrastive **Ca**ptioners
A method for image classification with transformer-based architecture.
See: section 2.4.2

CNN – **C**onvolutional **N**eural **N**etwork
An architecture of machine learning models that, utilizing principles of linear algebra, are capable of extracting features and identify patterns from data without any prior knowledge.

DL – **D**eep **L**earning
A subset of ML that utilizes the complex multilayered neural networks.

Doc2Vec – **Doc**ument **to Vec**tor
An algorithm to generate feature vectors from text documents.
See: section 2.4.9

ED – **E**uclidean **D**istance
A measure to quantity the similarity of the data.
See: section 2.3.2

FN – **F**alse **N**egative
A number of known data incorrectly recognized as an outlier,
See: section 2.2.3

FP – **F**alse **P**ositive
A number of unknown data incorrectly recognized as an inlier.
See: section 2.2.3

IAOF – **I**nterquartile **A**ngle-based **O**utlier **F**actor
A measure to quantity the similarity of the data.

ID – **I**n-**D**istribution
The typical data that are not likely to be outliers.

IRWD – **I**ntegrated **R**ank **W**eighted **D**epth
A measure to quantity the similarity of the data.
See: section 2.3.3

kNN – **k**-**N**earest **N**eighbors
An algorithm for identifying closest neighbor points located in space.
See: section 2.3.4

LOF – **L**ocal **O**utlier **F**actor
A measure to quantity the similarity of the data.
See: section 2.3.5

MD – **M**ahalanobis **D**istance
A measure to quantity the similarity of the data.
See: section 2.3.6

MDP – **M**ahalanobis **D**istance with **P**ooled covariance matrix
A measure to quantity the similarity of the data.
See: section 2.3.6

ML – **M**achine **L**earning
A branch of Computer Science focused on imitating the way that humans learn, to produce tools to recognize and classify the data.

MVN – **M**ulti**v**ariate **N**ormal distribution
A generalized normal/Gaussian distribution for multiple dimensions.

NLP – **N**atural **L**anguage **P**rocessing
A branch of Machine Learning focused on analyzing text data.

NN – **N**eural **N**etwork
A ML model that simulates the operation of biological neurons.

OOD – **O**ut-**o**f-**D**istribution
The abnormal data that are likely outliers for a given distribution.

OF – **O**utlier **F**actor
A measure of similarity used by OOD detector (general term).
See: section 2.3

PCA – **P**rincipal **C**omponent **A**nalysis
A method to reduce the dimensionality of feature vectors.

ResNet – **Res**idual **Net**works
A method for image classification that utilizes CNN architecture.
See: section 2.4.6

RP – **R**andom **P**rojection
A method to reduce the dimensionality of feature vectors.

SED – **S**tandardized **E**uclidean **D**istance
A measure to quantity the similarity of the data.
See: section 2.3.7

TF-IDF – **T**erm **F**requency – **I**nverse **D**ocument **F**requency
A way to represent text documents as feature vectors.

TN – **T**rue **N**egative
A number of unknown data correctly recognized as an outlier,
See: section 2.2.3

TP – **T**rue **P**ositive
A number of known data correctly recognized as an inlier.
See: section 2.2.3

ViT – **Vi**sion **T**ransformer
A method for image classification with transformer-based architecture.
See: section 2.4.7

Appendix B

Source code

The complete source code related to the conducted scientific research and the final preparation of the following dissertation is publicly available in the GitHub repositories:

- `https://github.com/sdatko/PhDatko`
- `https://github.com/sdatko/PyOpenSet`

B.1 PhDatko

This repository contains source files of the tooling used during the research and work on the final dissertation, including the LaTeX source files of this document, so all the conducted experiments can be repeated in case of any follow-up research is needed. It is divided into two major parts.

The `thesis/` directory contains the source files involved in building the current document. There is the `Makefile` file provided, so the dissertation can be compiled, provided that the LaTeX compiler available in system, using the following command:

```
make thesis
```

Inside the `research/` directory the source files of developed application are located. The application, written in the Python programming language, utilizes the Streamlit library to provide interactive viewer interface in web browser for analyzing the results of all conducted experiments. The results are calculated on demand if no cache is available. It is possible to use the provided `tox` environment to conveniently create the Python virtual environment with all required dependencies and run the developed application:

```
tox -e streamlit
```

```python
#!/usr/bin/env python3

from openset.data.generator import ClusterGenerator


def main():
    generator = ClusterGenerator()
    generator.reset(42)

    data = generator.mvn(samples=60, dimension=30,
                         location=4.0, scale=1.0,
                         n_features=0.75, n_correlated=0.5, covariance=0.25)

    print(data)


if __name__ == '__main__':
    main()
```

Listing B.1: Example usage of PyOpenSet library to generate data cluster

B.2 PyOpenSet

This repository contains the developed helper library for performing outlier detection / implementing open-set classification in high-dimensional data. It was written in Python programming language and consists of data generators, implemented outlierness measures and additional utilities, such as local runner for automated multiprocessing and general-purpose mechanism for persistent caching of function calls in SQLite database.

The library can be installed in local system using the following command:

```
pip install git+https://github.com/sdatko/PyOpenSet.git@master
```

Listing B.1 illustrates how PyOpenSet library can be utilized in a Python script to generate a data cluster containing 60 samples of dimension 30, each with 75% of features centered around the location 4.0 (mean) with a spread of 1.0 (standard deviation), having 50% of features correlated with a strength of 0.25 (covariance).

The **`examples/`** directory in the code repository provides additional usage examples for one's reference.

Appendix C

Selected personal achievements

This chapter covers the author's personal background.

C.1 List of scientific publications

Detailed list can be found in the database of the Wrocław University of Science and Technology: `https://dona.pwr.edu.pl/szukaj/default.aspx?nrewid=600305`

- Szymon Datko, Kamil Szyc, Tomasz Walkowiak, Henryk Maciejewski, *"How Characteristics of High-Dimensional Representations in Image and Text Recognition Impact the Performance of OOD Detectors"*, 2024.
  Forthcoming: under review.
- Szymon Datko, Henryk Maciejewski, Tomasz Walkowiak, *"Measures of Outlierness in High-Dimensional Data under Correlation of Features - with Application for Open-Set Classification"*, proceedings of the Seventeenth International Conference on Dependability of Computer Systems, DepCoS-RELCOMEX, 2022.
  DOI: `https://doi.org/10.1007/978-3-031-06746-4_3`
- Tomasz Walkowiak, Szymon Datko, Henryk Maciejewski, *"Utilizing Local Outlier Factor for Open-Set Classification in High-Dimensional Data - Case Study Applied for Text Documents"*, Intelligent Systems and Applications: proceedings of the 2019 Intelligent Systems Conference, IntelliSys, 2019.
  DOI: `https://doi.org/10.1007/978-3-030-29516-5_33`
- Tomasz Walkowiak, Szymon Datko, Henryk Maciejewski, *"Distance metrics in Open-Set Classification of Text Documents by Local Outlier Factor and Doc2Vec"*, Advances and trends in artificial intelligence: from theory to practice: 32nd

International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE, 2019.
DOI: https://doi.org/10.1007/978-3-030-22999-3_10

- Tomasz Walkowiak, Szymon Datko, Henryk Maciejewski, *"Low-dimensional classification of text documents"*, Engineering in dependability of computer systems and networks: proceedings of the Fourteenth International Conference on Dependability of Computer Systems, DepCoS-RELCOMEX, 2019.
  DOI: https://doi.org/10.1007/978-3-030-19501-4_53

- Tomasz Walkowiak, Szymon Datko, Henryk Maciejewski, *"Open Set Subject Classification of Text Documents in Polish by Doc-to-Vec and Local Outlier Factor"*, Artificial Intelligence and Soft Computing: 18th International Conference, ICAISC, 2019.
  DOI: https://doi.org/10.1007/978-3-030-20915-5_41

- Tomasz Walkowiak, Szymon Datko, Henryk Maciejewski, *"Reduction of dimensionality of feature vectors in subject classification of text documents"*, Reliability and statistics in transportation and communication: selected papers from the 18th International Conference on Reliability and Statistics in Transportation and Communication, RelStat, 2018.
  DOI: https://doi.org/10.1007/978-3-030-12450-2_15

- Tomasz Walkowiak, Szymon Datko, Henryk Maciejewski, *"Bag-of-Words, Bag-of-Topics and Word-to-Vec Based Subject Classification of Text Documents in Polish - A Comparative Study"*, Contemporary complex systems and their dependability: proceedings of the Thirteenth International Conference on Dependability and Complex Systems, DepCoS-RELCOMEX, 2018.
  DOI: https://doi.org/10.1007/978-3-319-91446-6_49

- Tomasz Walkowiak, Szymon Datko, Henryk Maciejewski, *"Feature Extraction in Subject Classification of Text Documents in Polish"*, Artificial Intelligence and Soft Computing: 17th International Conference, ICAISC, 2018.
  DOI: https://doi.org/10.1007/978-3-319-91262-2_40

- Tomasz Walkowiak, Szymon Datko, Henryk Maciejewski, *"Algorithm Based on Modied Angle-Based Outlier Factor for Open-Set Classification of Text Documents"*, Applied Stochastic Models in Business and Industry, ASMBI, 2018.
  DOI: https://doi.org/10.1002/asmb.2388

C.2 List of conference speeches

- Szymon Datko, Adrian Fusco Arnejo, *"Dashboards as a Code: managing Grafana with Jsonnet"*, OpenInfra Day Germany, May 2024, Berlin, Germany.
  Resources: `https://github.com/adrianfusco/openinfra2024-dashboard-as-a-code`

- Szymon Datko, Ignacio Horcada Bernal, *"Debugging Zuul jobs – now easier than ever, with Autoholds feature"*, OpenInfra Summit Vancouver '23, June 2023, Vancouver, Canada.
  Recording: `https://www.youtube.com/watch?v=_GEaQGhZd9Y`

- Arie Bregman, Szymon Datko, *"Combining Ansible and Terraform for CI – better together love story based on OVN-CI project"*, Virtual Open Infrastructure Summit, November 2020.
  Recording: `https://www.youtube.com/watch?v=6D13rG0iawI`

- Szymon Datko, Roman Dobosz, *"Zuul, the Third – Throws Away Any Dirt! A quick-start introduction"*, OpenInfra Summit Shanghai 2019, November 2019, Shanghai, China.
  Recording: `https://www.youtube.com/watch?v=_viUYriGdPw`

- Szymon Datko, Roman Dobosz, *"Does your Jenkins speak Gerrit? Functional testing for your pipelines, JobDSL and more"*, OpenInfra Summit Shanghai 2019, November 2019, Shanghai, China.
  Recording: `https://www.youtube.com/watch?v=PmgIGnUrV5g`

- Szymon Datko, Tomasz Walkowiak, Henryk Maciejewski, *"Low-dimensional classification of text documents"*, 14th International Conference on Dependability of Computer Systems – DepCoS 2019, July 2019, Brunów Palace, Lwówek Śląski, Poland.

- Szymon Datko, *"Zuul trzeci – wyrzuca zły kod na śmieci"*, OpenInfra Days Poland 2019, June 2019, Kraków, Poland.

- Szymon Datko, Roman Dobosz, *"Testing Jenkins configuration changes – solidify your JCasC, Job DSL and Pipelines usage"*, OpenInfra Summit Denver 2019, May 2019, Denver, USA.
  Recording: `https://www.youtube.com/watch?v=nvgeXkE65ac`

- Piotr Bielak, Szymon Datko, *"Zuul v3 – more than a project gating system"*, OpenInfra Wrocław Meetup #11, March 2019, Wrocław, Poland.

- Piotr Bielak, Szymon Datko, *"From messy XML to wonderful YAML and pretty Job DSL – an in-Jenkins migration story"*, OpenStack Summit Berlin 2018, November 2018, Berlin, Germany.
  Recording: `https://www.youtube.com/watch?v=T7rD--ZOYRQ`

- Szymon Datko, *"Aktualizacja OpenStacka – świeży raport z pola bitwy"*, OpenStack Days Poland 2018, June 2018, Kraków, Poland.

- Szymon Datko, *"CDS - simple, scalable, powerful CI/CD solution"*, 14th Linux Session, May 2017, Wrocław University of Science and Technology, Wrocław, Poland.
  Recording: `https://www.youtube.com/watch?v=RneLKacYVC0`

- Szymon Datko, Henryk Maciejewski, *"Outlier Detection in High-Dimensional Data – Applied for Open-Set Text Classification"*, 13th Workshop on Stochastic Models, Statistics and Their Applications, February 2017, Humboldt-Universität zu Berlin, Berlin, Germany.

- Szymon Datko, *"Automate your life with Gitlab-CI"*, Student Session 2016, August 2016, CERN European Organization for Nuclear Research, Geneva, Switzerland.
  Recording: `https://cds.cern.ch/record/2206413`

C.3 Projects and grants

- Henryk Maciejewski (Principal Investigator), Tomasz Walkowiak (Co-Investigator), Szymon Datko (Auxiliary investigator), *Classification based on high-dimensional open-set data - with applications in Text Mining*, OPUS 11, National Science Centre, Poland (grant **2016/21/B/ST6/02159**).

C.4 Other achievements

- Supported organization and running of the *Konferencja Projektów Zespołowych* – an event for third-year students during which their team projects are presented to the audience of academic community members and invited industry representatives. It involved the coordination of contact between the companies and the University, registration of projects, preparing gifts and prizes for the conference participants, designing and ordering souvenir T-shirts, providing technical support during final projects presentations (8 editions, 2017–2024).
  URL: `https://kpz.pwr.edu.pl`

- Participated in the creation of a new course Machine Learning in Animations, including preparation of teaching materials, as part of the AI Tech project (2022).
- Assisted in the creation of a new specialization *Grafika i Systemy Multimedialne* (ang. *Graphics and Multimedia Systems*) in the field of Computer Science at Wrocław University of Science and Technology – preparation of new courses cards, exam questions and teaching materials for the first and second degrees of studies as part of the ZPR POWER project (2020).
- Prepared courses "Computer Graphics and Game Development Fundamentals" (2019) and "Introduction to DevOps and automation" (2020) for the TECHSummer international summer school, addressed to students from partner universities in India.
- Organized a series of meetings and training on Linux systems administration and server infrastructure management for the members of the Section for Information Technology of the Student Government at Wrocław University of Science and Technology – as part of the Red Hat Academy program (2022–2023).
- Participated in the *"PROJEKTOR"* voluntary work as part of the *"IT for SHE"* program – organization of classes popularizing Computer Science and related fields (STEM – Science, Technology, Engineering and Math) among children and teenagers during summer camps, organized at the major Henryk Dobrzański's Primary School in Bircza, Poland (2017) and at the W. Witos's Public Primary School in Borek Strzeliński, Poland (2019).
- Conducted in total over 2500 hours of classes for students, teaching topics such as: computer graphics, animations and simulations, design and programming of computer games, software processing of images, scripting in operating systems, acceleration of computations and diagnostics of digital circuits (2016-2024).
- Established an educational YouTube channel and prepared recorded introductions to laboratory classes for students, from topics of computer graphics and shell scripts programming (2021–2024).
  URL: `https://www.youtube.com/c/SzymonDatko/videos`

List of Figures

Bibliography

[1] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in ai safety, 2016.

[2] Y. Bengio, G. Hinton, A. Yao, D. Song, P. Abbeel, T. Darrell, Y. N. Harari, Y.-Q. Zhang, L. Xue, S. Shalev-Shwartz, G. Hadfield, J. Clune, T. Maharaj, F. Hutter, A. G. Baydin, S. McIlraith, Q. Gao, A. Acharya, D. Krueger, A. Dragan, P. Torr, S. Russell, D. Kahneman, J. Brauner, and S. Mindermann. Managing extreme ai risks amid rapid progress. Science, 384(6698):842–845, 2024.

[3] J. L. Bentley. Multidimensional binary search trees used for associative searching. Commun. ACM, 18(9):509–517, sep 1975.

[4] J. Bitterwolf, M. Mueller, and M. Hein. In or out? fixing imagenet out-of-distribution detection evaluation. In ICML, 2023.

[5] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information, 2017.

[6] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In ACM sigmod record, volume 29, pages 93–104. ACM, 2000.

[7] J. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In D. Touretzky, editor, Advances in Neural Information Processing Systems, volume 2. Morgan-Kaufmann, 1989.

[8] R. A. Brown. Building a balanced k-d tree in $o(kn \log n)$ time. Journal of Computer Graphics Techniques (JCGT), 4(1):50–68, March 2015.

[9] I. Casanueva, T. Temcinas, D. Gerz, M. Henderson, and I. Vulic. Efficient intent detection with dual sentence encoders. In Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020, mar 2020. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

[10] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. ACM Comput. Surv., 41(3), jul 2009.

[11] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild, 2013.

[12] P. Colombo, E. Dadalto, G. Staerman, N. Noiry, and P. Piantanida. Beyond mahalanobis distance for textual ood detection. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, Advances in Neural Information Processing Systems, volume 35, pages 17744–17759. Curran Associates, Inc., 2022.

[13] J. C. Costa, T. Roxo, H. Proença, and P. R. M. Inácio. How deep learning sees the world: A survey on adversarial attacks & defenses. IEEE Access, 12:61113–61136, 2024.

[14] S. Datko, H. Maciejewski, and T. Walkowiak. Measures of outlierness in high-dimensional data under correlation of features – with application for open-set classification. In W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, editors, New Advances in Dependability of Networks and Systems, pages 22–31, Cham, 2022. Springer International Publishing.

[15] S. Datko, K. Szyc, T. Walkowiak, and H. Maciejewski. How characteristics of high-dimensional representations in image and text recognition impact the performance of ood detectors. Forthcoming (under review), 2024.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[19] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou. The faiss library, 2024.

[20] X. Du, Z. Wang, M. Cai, and Y. Li. Vos: Learning what you don't know by virtual outlier synthesis, 2022.

[21] S. Fort, J. Ren, and B. Lakshminarayanan. Exploring the limits of out-of-distribution detection. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, Advances in Neural Information Processing Systems, volume 34, pages 7068–7081. Curran Associates, Inc., 2021.

[22] C. Geng, S.-J. Huang, and S. Chen. Recent advances in open set recognition: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(10):3614–3631, Oct. 2021.

[23] D. Goldberg. What every computer scientist should know about floating-point arithmetic. ACM Comput. Surv., 23(1):5–48, mar 1991.

[24] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning: data mining, inference and prediction. Springer, 2 edition, 2009.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[26] D. Hendrycks, S. Basart, M. Mazeika, A. Zou, J. Kwon, M. Mostajabi, J. Steinhardt, and D. Song. Scaling out-of-distribution detection for real-world settings, 2022.

[27] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization, 2021.

[28] D. Hendrycks, N. Carlini, J. Schulman, and J. Steinhardt. Unsolved problems in ml safety, 2022.

[29] D. Hendrycks, M. Mazeika, and T. Dietterich. Deep anomaly detection with outlier exposure, 2019.

[30] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural adversarial examples. CVPR, 2021.

[31] N. J. Higham. Analysis of the Cholesky decomposition of a semi-definite matrix. In Reliable Numerical Commputation. Oxford University Press, 09 1990.

[32] V. Hodge and J. Austin. A survey of outlier detection methodologies. Artificial Intelligence Review, 22:85–126, 10 2004.

[33] G. V. Horn, O. M. Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. The inaturalist species classification and detection dataset, 2018.

[34] A. Hotho, A. Nürnberger, and G. Paaß. A brief survey of text mining. Journal for Language Technology and Computational Linguistics, 20(1):19–62, 2005.

[35] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[36] G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt. Openclip, July 2021.

[37] R. Johnson and D. Wichern. Applied Multivariate Statistical Analysis. Applied Multivariate Statistical Analysis. Pearson Prentice Hall, 2007.

[38] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification, 2016.

[39] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(1):117–128, 2011.

[40] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 444–452. ACM, 2008.

[41] K. Lang. Newsweeder: Learning to filter netnews. In A. Prieditis and S. Russell, editors, Machine Learning Proceedings 1995, pages 331–339. Morgan Kaufmann, San Francisco (CA), 1995.

[42] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents, 2014.

[43] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, page 7167–7177, Red Hook, NY, USA, 2018. Curran Associates Inc.

[44] T. Liu, A. W. Moore, and A. Gray. New algorithms for efficient high-dimensional nonparametric classification. Journal of Machine Learning Research, 7(41):1135–1158, 2006.

[45] W. Liu, X. Wang, J. Owens, and Y. Li. Energy-based out-of-distribution detection. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 21464–21475. Curran Associates, Inc., 2020.

[46] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s, 2022.

[47] P. C. Mahalanobis. On the generalized distance in statistics. In Proceedings of the National Institute of Science of India, volume 2, pages 49–55, 1936.

[48] Microprocessor Standards Committee, Floating-Point Working Group and others. Ieee standard for floating-point arithmetic. IEEE Std 754-2019 (Revision of IEEE 754-2008), pages 1–84, 2019.

[49] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[50] S. M. Omohundro. Five balltree construction algorithms. Technical Report TR-89-063, International Computer Science Institute, December 1989.

[51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[52] A. Podolskiy, D. Lipin, A. Bout, E. Artemova, and I. Piontkovskaya. Revisiting mahalanobis distance for transformer-based out-of-domain detection, 2022.

[53] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[54] K. Ramsay, S. Durocher, and A. Leblanc. Integrated rank-weighted depth. Journal of Multivariate Analysis, 173, 02 2019.

[55] E. Raninen, D. E. Tyler, and E. Ollila. Linear pooling of sample covariance matrices. IEEE Transactions on Signal Processing, 70:659–672, 2022.

[56] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. A. DePristo, J. V. Dillon, and B. Lakshminarayanan. Likelihood ratios for out-of-distribution detection, 2019.

[57] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.

[58] P. Rousseeuw. Least median of squares regression. Journal of The American Statistical Association - J AMER STATIST ASSN, 79:871–880, 12 1984.

[59] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3):211–252, Dec. 2015. Publisher Copyright: © 2015, Springer Science+Business Media New York.

[60] A. Singh, A. Yadav, and A. Rana. K-means with three different distance metrics. International Journal of Computer Applications, 67:13–17, 04 2013.

[61] J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation, 23(5):828–841, Oct. 2019.

[62] Y. Sun, Y. Ming, X. Zhu, and Y. Li. Out-of-distribution detection with deep nearest neighbors, 2022.

[63] K. Szyc, T. Walkowiak, and H. Maciejewski. Why out-of-distribution detection experiments are not reliable - subtle experimental details muddle the OOD detector rankings. In R. J. Evans and I. Shpitser, editors, Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence, volume 216 of Proceedings of Machine Learning Research, pages 2078–2088. PMLR, 31 Jul–04 Aug 2023.

[64] J. Tack, S. Mo, J. Jeong, and J. Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 11839–11852. Curran Associates, Inc., 2020.

[65] F. Tajwar, A. Kumar, S. M. Xie, and P. Liang. No true state-of-the-art? ood detection methods are inconsistent across datasets, 2021.

[66] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[67] J. Tukey. Exploratory Data Analysis. Number t. 2 in Addison-Wesley series in behavioral science. Addison-Wesley Publishing Company, 1977.

[68] V. Vapnik. The Nature of Statistical Learning Theory. Information Science and Statistics. Springer New York, 1999.

[69] D. S. Vijayarani, M. J. Ilamathi, and M. N. S. Nithya. Preprocessing techniques for text mining - an overview. International Journal of Computer Science & Communication Networks, 5:7–16, 02 2015.

[70] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17:261–272, 2020.

[71] T. Walkowiak, S. Datko, and H. Maciejewski. Algorithm based on modified angle-based outlier factor for open-set classification of text documents. Applied Stochastic Models in Business and Industry, 34(5):718–729, 2018.

[72] T. Walkowiak, S. Datko, and H. Maciejewski. Feature extraction in subject classification of text documents in polish. In L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, and J. M. Zurada, editors, Artificial Intelligence and Soft Computing, pages 445–452, Cham, 2018. Springer International Publishing.

[73] T. Walkowiak, S. Datko, and H. Maciejewski. Bag-of-words, bag-of-topics and word-to-vec based subject classification of text documents in polish - a comparative study. In W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, editors, Contemporary Complex Systems and Their Dependability, pages 526–535, Cham, 2019. Springer International Publishing.

[74] T. Walkowiak, S. Datko, and H. Maciejewski. Distance metrics in open-set classification of text documents by local outlier factor and doc2vec. In F. Wotawa, G. Friedrich, I. Pill, R. Koitz-Hristov, and M. Ali, editors, Advances and Trends in Artificial Intelligence. From Theory to Practice, pages 102–109, Cham, 2019. Springer International Publishing.

[75] T. Walkowiak, S. Datko, and H. Maciejewski. Open set subject classification of text documents in polish by doc-to-vec and local outlier factor. In L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, and J. M. Zurada, editors, Artificial Intelligence and Soft Computing, pages 455–463, Cham, 2019. Springer International Publishing.

[76] T. Walkowiak, S. Datko, and H. Maciejewski. Reduction of dimensionality of feature vectors in subject classification of text documents. In I. Kabashkin, I. Yatskiv (Jackiva), and O. Prentkovskis, editors, Reliability and Statistics in Transportation and Communication, pages 159–167, Cham, 2019. Springer International Publishing.

[77] T. Walkowiak, S. Datko, and H. Maciejewski. Low-dimensional classification of text documents. In W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, editors, Engineering in Dependability of Computer Systems and Networks, pages 534–543, Cham, 2020. Springer International Publishing.

[78] T. Walkowiak, S. Datko, and H. Maciejewski. Utilizing local outlier factor for open-set classification in high- dimensional data - case study applied for text documents. In Y. Bi, R. Bhatia, and S. Kapoor, editors, Intelligent Systems and Applications, pages 408–418, Cham, 2020. Springer International Publishing.

[79] H. Wang, Z. Li, L. Feng, and W. Zhang. Vim: Out-of-distribution with virtual-logit matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.

[80] L. Wasserman. All of statistics : a concise course in statistical inference. Springer, New York, 2010.

[81] J. Winkens, R. Bunel, A. G. Roy, R. Stanforth, V. Natarajan, J. R. Ledsam, P. MacWilliams, P. Kohli, A. Karthikesalingam, S. Kohl, T. Cemgil, S. M. A. Eslami, and O. Ronneberger. Contrastive training for improved out-of-distribution detection, 2020.

[82] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In 2010 IEEE computer society conference on computer vision and pattern recognition, pages 3485–3492. IEEE, 2010.

[83] J. Yang, P. Wang, D. Zou, Z. Zhou, K. Ding, W. Peng, H. Wang, G. Chen, B. Li, Y. Sun, X. Du, K. Zhou, W. Zhang, D. Hendrycks, Y. Li, and Z. Liu. Openood: Benchmarking generalized out-of-distribution detection, 2022.

[84] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu. Coca: Contrastive captioners are image-text foundation models, 2022.

[85] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

[86] W. Zhou, F. Liu, and M. Chen. Contrastive out-of-distribution detection for pretrained transformers. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1100–1111, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.