An Examination of Meta-Learning for Algorithm Selection in Unsupervised Anomaly Detection

Malgorzata Gutowska, MSc

Supervised by
Dr Andrew McCarren and Dr Suzanne Little

DCU
Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

A thesis presented for the degree of Doctor of Philosophy

School of Computing
Dublin City University

January 2024

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

Mgutowska

Malgorzata Gutowska

ID No.: 17212089

Date: January 10th, 2024

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Dr Andrew McCarren and Dr Suzanne Little, for their tireless support, guidance, and mentorship throughout my PhD journey. They provided critical insights that helped shape my research, and consistently challenged me to think more deeply about my work. Their expertise and constructive feedback have been instrumental in shaping this research and aiding my intellectual growth. The collaborative environment they fostered allowed me to explore complex questions, troubleshoot challenges, and ultimately, produce work that I am proud of. I feel incredibly fortunate to have had the opportunity to learn from and work alongside such distinguished academics.

I am also thankful to our CRT Program Manager, Janet Choi, and CRT Officer, Angela Lally, for their incredible organisational efforts in arranging meetings and events that were crucial for my research and professional development. Their willingness to go above and beyond in encouraging not only academic but also personal growth has been a source of strength and inspiration. I am truly thankful for their role in my scholarly journey. Likewise, I wish to thank my colleagues and collaborators, particularly Dr Michael Scriney, for their assistance with various technical matters. Their collaborative efforts have helped me overcome obstacles and boost the quality of my work.

Lastly, this work would not have been possible without the financial support from Science Foundation Ireland (SFI), whose funding provided the resources required to conduct this research. To everyone who has played a part in this academic endeavor, I offer my sincerest thanks.

Dedication

To my steadfast anchor and beloved husband,

Your consistent encouragement and support were crucial in finishing this project. Your understanding enabled me to devote the time and energy required to reach this milestone. Your contribution, while less visible, has been equally essential. Thank you for being a rock during this difficult journey.

Contents

List of Figures

List of Tables

List of Publications

1. M. Gutowska, M. Scriney, and A. McCarren, "Identifying extra-terrestrial intelligence using machine learning," in Proceedings for the 27th Irish Conference on Artificial Intelligence and Cognitive Science (AICS2019), 2563 CEUR-WS, pp. 293-304, 2020.

2. M. Gutowska, S. Little and A. McCarren, "Constructing a Meta-Learner for Unsupervised Anomaly Detection," in *IEEE Access*, vol. 11, pp. 45815-45825, 2023, doi: 10.1109/ACCESS.2023.3274113.

3. M. Gutowska, S. Little and A. McCarren, "Model-Agnostic Framework for Evaluating the Reliability of Meta-Learner Algorithm Suggestions for Unsupervised Anomaly Detection," [Manuscript submitted for publication], 2023.

An Examination of Meta-Learning for Algorithm Selection in Unsupervised Anomaly Detection

Malgorzata Gutowska

Abstract

Detecting anomalies is crucial for a range of applications, including network security and healthcare. A primary challenge in anomaly detection (AD) is its unsupervised nature, prevalent in most real-world scenarios. Despite a multitude of existing AD algorithms, no single approach succeeds across all anomaly detection tasks. While the Algorithm Selection Problem (ASP) has been extensively studied in supervised learning through meta-learning and AutoML techniques, it has received little attention in the unsupervised domain. The absence of efficient strategies for algorithm selection and evaluation is a matter that requires attention. This dissertation employs meta-learning techniques tailored to unsupervised anomaly detection in an effort to bridge this gap.

The study introduces a new meta-learner designed to select the most suitable unsupervised AD algorithm for unlabelled datasets. The proposed meta-learner outperforms the current state-of-the-art solution. Furthermore, this research includes an analysis of the individual components of the meta-learner, such as the meta-model, meta-features, and the base set of AD algorithms. It reveals that the design of the meta-model is essential for effective meta-learning. In evaluating the meta-learner's recommendations, the research provides a framework for assessing both the risk of inaccurate responses and the potential errors in individual predictions. Moreover, this study employs a comprehensive collection of over 10,000 datasets, providing a robust foundation for its findings.

This research addresses a crucial gap in existing literature by offering a systematic methodology for algorithm selection in unsupervised AD, a particularly urgent problem given the exponential growth of data and the corresponding demand for reliable AD mechanisms. As such, this work enhances data management capabilities in increasingly data-saturated environments.

Chapter 1

Introduction

This thesis presents a meta-learning approach to addressing the Algorithm Selection Problem (ASP) in the context of unsupervised Anomaly Detection (AD). Detecting anomalies is essential in almost every industry or domain due to the abundance of collected data. The primary challenge in effective anomaly detection is its intrinsic unsupervised character, which is common for most practical scenarios. Despite its importance, the field of systematic methods for unsupervised anomaly detection remains largely unexplored, resulting in a lack of efficient tools for selecting the best algorithm for a specific task and assessing the trustworthiness of such a selection. This thesis aims to address the existing research gap.

1.1 Anomaly Detection

Anomalies are instances of data that deviate unexpectedly from other instances. They are also referred to as outliers in the context of data analytics. The detection of these outlier data points is gaining increasing attention in a variety of business sectors and other everyday aspects, owing primarily to the growing number of systems that collect and utilise data generated by a variety of daily activities.

Outlier detection has evolved throughout time in response to changing needs. Historically, outliers were most commonly identified by data cleaning, a primarily manual process. Many data analysis techniques assume that the input data contains only inliers;

thus, removing outliers is a critical data preprocessing step. As data-driven systems become more prevalent in everyday life, the necessity for AD grows. Detecting anomalies in sophisticated systems not only helps them to run smoothly, but in many domains, the identified anomaly is a true value by itself. These outliers may contain critical information about the system, the data, or the environment in which the data was collected. An outlier can be a potentially harmful event that should be avoided, or an early indicator of a new trend that should not go unnoticed. Insurance, banking, monitoring network traffic, or health are a few examples of the broad range of domains where anomaly detection is critical. The existing literature gives an extensive review of applications for anomaly detection (Campos et al. 2016; Chalapathy et al. 2019; Ruff et al. 2021; H. Wang et al. 2019).

Detecting anomalies from a series of observations can be considered as a type of classification task in which classes are significantly imbalanced. However, there are other factors that make this type of problem more distinct. Unlike in many classification problems, the most common techniques in AD are based on either unsupervised or semi-supervised learning rather than supervised learning. This is because in real AD problems, labelled anomalous data is rarely available. In unsupervised learning, the identification of anomalies is purely based on the patterns within the data, which are identifiable to the algorithm, whereas, in semi-supervised learning, the model is trained in advance and only exposed to normal data during training (Chalapathy et al. 2019; Goldstein and Uchida 2016).

Typically, AD algorithms seek to estimate the likelihood of every data point being an anomaly in an unsupervised manner (Goldstein and Uchida 2016), considering the anomaly characteristics. These algorithms return anomaly scores (usually not normalised), where a higher score indicates an increased likelihood that a given data point is an anomaly (Chandola et al. 2009; Goldstein and Uchida 2016). Based solely on the anomaly scores it is not possible to measure an algorithm's performance in a form of a single metric such as accuracy, recall or precision. A detailed study of the results is usually necessary to assess the algorithm performance on a new task.

1.2 Algorithm Selection Problem

Numerous techniques have been proposed for AD, but multiple studies support the consensus that no single technique is optimal for all AD problems (Campos et al. 2016; H. Wang et al. 2019). The problem of selecting the best algorithm for a given task is referred to as the *Algorithm Selection Problem* (ASP) (Ali et al. 2017; Bischl et al. 2016; I. Khan et al. 2020), a challenge that is prevalent in both supervised and unsupervised contexts.

Traditional solutions to the ASP, such as trial-and-error or consultation with domain experts, are not always practical. The trial-and-error approach is often time-consuming and lacks guarantees of finding the optimal solution. On the other hand, engaging human expertise can be prohibitively costly and still offers no assurance of success.

To address these shortcomings, automated selection of the best performing algorithm has been regarded to be a more effective strategy. The concept of building a model capable of learning from previous evaluations to predict the performance of algorithms on new tasks, referred to as meta-learning, is well-established within the ASP context (Hutter et al. 2019; I. Khan et al. 2020; Lemke et al. 2015; Muñoz et al. 2015). Meta-learning, as a key component of Automated Machine Learning (AutoML), is a fast-growing area within the machine learning discipline (Hutter et al. 2019).

One of the prevalent strategies in meta-learning involves analysing the characteristics of datasets, which can then be mapped to the performance of specific algorithm configurations based on historical evaluations (Hutter et al. 2019; Smith-Miles 2009). This idea of mapping the task attributes to algorithm performance was first conceptualized in Rice's seminal paper (Rice 1976) and has since been expanded upon by researchers such as Smith-Miles 2009. This expanded framework for algorithm selection is depicted in Figure 1.1.

There are four essential components of the framework (Smith-Miles 2009):

- problem space P – a set of dataset instances of a problem,
- feature space F – a set of characteristics (meta-features) generated from each dataset instance x,

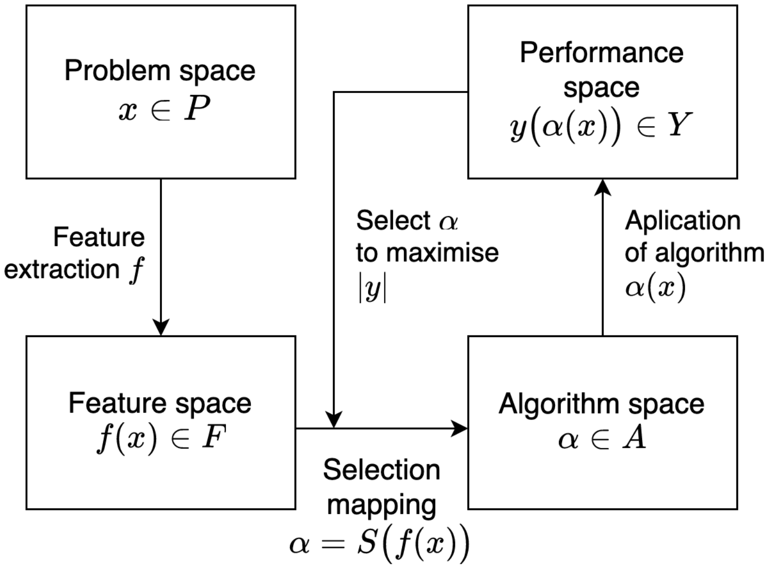


Figure 1.1: Rice's framework, as presented by Smith-Miles 2009.

- algorithm space A – a set of algorithms (possibly including variations incorporating hyperparameters),
- performance space Y – a set of performance metrics of the algorithms from A evaluated over the problem space P.

The ASP can be defined as "For a given problem instance $x \in P$ with features $f(x) \in F$, find the selection mapping $S(f(x))$ into algorithm space A, such that the selected algorithm $\alpha \in A$ maximises the performance mapping $y(\alpha(x)) \in Y$" (Smith-Miles 2009).

The ASP is even more challenging when faced with an unsupervised task. Applying a trial-and-error approach or maximisation of performance mapping is problematic without a single performance metric. A meta-learner that indicates the best algorithm for a new unsupervised task is thus a necessary tool, particularly in scenarios with high up-front uncertainty and high variability.

Currently, the ASP for unsupervised settings remains largely unexplored. A few studies have examined the meta-features' potential in AD scenarios (Campos et al. 2016; Kandanaarachchi et al. 2020; Kotlar et al. 2021), but their limitations are mainly due to the required knowledge of data labels or its structure.

The concurrent approach to algorithm selection for handling unsupervised AD tasks often involves using an ensemble technique or a "voting" approach to identify anomalies (Le Clei et al. 2022; Papastefanopoulos et al. 2021). However, the scalability of this approach is limited. Every new dataset or an inclusion of another algorithm requires multiple evaluations. This can be particularly challenging when timely responses are crucial, such as in near-real-time anomaly detection. On the contrary, utilising historical evaluations and applying knowledge transfer via meta-learning has the potential to offer an efficient and effective method for managing unsupervised AD tasks.

MetaOD (Meta-learning-based Outlier Detection), the approach presented by Zhao, Rossi, et al. 2021, represents the current state-of-the-art solution in the ASP problem for unsupervised AD. It leverages the knowledge from previous algorithm evaluations to obtain the recommended algorithm for a new task. MetaOD utilises a base set of eight "classic" AD algorithms, combined with various hyperparameters, to create 302 distinct models. The training of the original MetaOD meta-learner was based on 162 AD benchmark datasets, characterised by 200 meta-features. The approach is inspired by recommender systems and employs collaborative filtering with a matrix factorisation technique at its core. Further details on this approach are provided in Chapters 2 and 4.

1.3 Problem Statement and Research Questions

This section discusses the open research areas in the field of ASP for unsupervised AD and encapsulates the research questions. It also summarises the contributions made by the current study to the explored subject.

Research Question 1. Meta-learner for unsupervised AD

As mentioned in Section 1.1, the unsupervised nature of AD introduces a range of challenges rarely encountered in supervised tasks. Evaluating a specific AD method is challenging due to the inability to apply straightforward performance metrics. Such evaluations typically require significant time, effort, and often expert supervision. Moreover, there is an established consensus in the field, supported by multiple studies (Campos et al. 2016; Emmott et al. 2015; Goldstein and Uchida 2016; Kandanaarachchi et al. 2020;

Zhao, Rossi, et al. 2021), that no single algorithm consistently outperforms others for all AD tasks. This phenomenon is not exclusive to AD and is a well-known concept called the "No Free Lunch" theorem (Wolpert et al. 1997). As highlighted in Section 1.2, approaches that employ ensemble techniques fall short in delivering time-efficient solutions or practical applications when a larger pool of algorithms is deemed essential. Automated Machine Learning (AutoML), a field that guides users in algorithm and hyperparameter selection, is primarily concentrated on providing solutions for supervised problems. This highlights a critical gap in the field: the need for tools that can recommend appropriate unsupervised AD techniques for an unseen task with unknown labels. The above, in conjunction with the current study's preliminary investigation, motivates the first research question.

RQ1 Can an efficient meta-learner for unsupervised anomaly detection recommend the best algorithm for an unseen and unlabelled dataset?

This question is addressed in Chapter 4, Sections 4.1 and 4.2.

Research Question 2. Contribution of meta-learner's components to its success

The construction of a meta-learner involves numerous design decisions. Firstly, it is essential to describe the datasets in a manner that is independent of ground-truth labels and also ensures that potential anomalies within the dataset are captured by the meta-features. Secondly, the meta-learner needs to be trained on existing and known problems, necessitating the selection of suitable base AD algorithms for this training process. The choices made in these areas can significantly influence the final performance of the meta-learner. Finally, the meta-learner's design and architecture are inevitably significant factors influencing its overall effectiveness. However, existing research studies often focus on comparing the end results with a baseline or state-of-the-art solutions, neglecting to analyze the underlying components or factors that contribute to the success of a particular solution. This leads to the second research question of this study.

RQ2 Which components and design decisions of the meta-learner influence its overall performance?

This study seeks to address the *RQ2* in Chapter 4, Section 4.3.

Research Question 3. Local reliability and risk

The development of a meta-learner and the evaluation of factors influencing its performance represent only one aspect of the overall solution. A subsequent question that arises is regarding the quality of the recommendations provided by the meta-learner. For unsupervised tasks, the assessment of the recommended algorithm's efficacy remains a challenging task due to the lack of a straightforward performance metric. Evaluating the aggregate performance of a meta-learner can be of limited utility, given that individual datasets reside in a highly non-linear space. As a result, the reliability of a meta-learner's response in one area may differ significantly from that in another area. The assessment of the reliability of individual responses becomes particularly crucial, as a single point in the meta-feature space corresponds to an entire AD problem with unique characteristics. This leads to the next research question of this study.

RQ3 Can the reliability of individual meta-learner responses be evaluated and high-risk areas within the meta-feature space be identified?

The research addressing this question is discussed in Chapter 5.

Research Question 4. Datasets

One key consideration in conducting a meta-analysis leading to the creation of a meta-learner, and the subsequent study of its characteristics, is the need for a large and varied collection of datasets. To date, research addressing the algorithm selection problem for unsupervised tasks has typically relied on a limited scope of datasets to validate their hypotheses. This observation prompts the final research question that this thesis seeks to explore:

RQ4 How does utilising a large set of benchmark datasets influence the findings and conclusions of a meta-learner investigation?

As this work makes use of the largest publicly available collection of AD benchmark datasets, the implications of its size are discussed throughout the thesis and through all presented experiments. In addition, Section 3.1 examines the chosen dataset collection in detail and explores its properties.

1.3.1 Contributions

In an effort to address the discussed research questions, this study offers the following contributions:

- Creation of an alternative meta-learner superior to the state-of-the-art solution,
- Identifying the elements with the highest contribution to the meta-learner performance,
- Risk assessment strategy of incorrect individual meta-learner responses,
- Meta-learning experiments performed on a large set of variable AD benchmark datasets allowing for deeper insights and understanding of the problem.

1.4 Thesis Outline

The remainder of this thesis is organized as follows: Chapter 2 provides a literature review encompassing the areas of anomaly detection, AutoML, the algorithm selection problem, and local evaluation of machine learning models. Chapter 3 introduces the datasets utilised in the experiments performed in this research, the selected metrics for AD, and an overview of the preliminary experiments conducted. In Chapter 4, the proposed meta-learner is described, detailing its evaluation and characteristics. Chapter 5 presents a proposed framework for risk assessment and local reliability. Lastly, Chapter 6 concludes the thesis by summarizing the key findings.

Chapter 2

Related Work

This section starts by showing the literature on Anomaly Detection (AD) and its relevant key aspects (Section 2.1); it later presents the literature on AutoML in relation to both areas – supervised (Section 2.2) and unsupervised (Section 2.3). In the final section, the research on local evaluation of machine learning models is discussed, especially in the context of regression models, as it is a foundation to the research presented in Chapter 5 (Section 2.4).

2.1 Anomaly Detection

2.1.1 Defining an Anomaly

One of the first descriptions of an outlier was given in 1969 by Grubbs: "An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs" (Grubbs 1969). Another highly cited definition of an anomaly has been given by Hawkins as "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism" (Hawkins 1980). Ruff et al. 2021 provide a formal definition of an anomaly based on the probability theory. They define the set of anomalies $\mathcal{A}$ as:

$$\mathcal{A} = \left\{ x \in \mathcal{X} \mid p^{+}(x) \leq \tau \right\}, \tau \geq 0 \tag{2.1}$$

where x is a data point in the data space $\mathcal{X}$, $p^{+}(x)$ is a density probability function and τ is a "sufficiently small" probability threshold.

The above definitions, as well as multiple other attempts made throughout the literature to define an anomaly, allow for the inference of the following high-level features, which are applicable in most scenarios (Goldstein and Uchida 2016):

- anomalies differ significantly from other data points,
- they are rare compared to normal instances.

In addition to capturing essential real-world qualities, the above characteristics have been the determining factors in the generation of synthetic datasets for anomaly detection purposes (Emmott et al. 2013, 2015; Goldstein and Uchida 2016).

2.1.2 Classification of Anomalies

Another challenge that goes hand in hand with the lack of a widely accepted definition of anomaly is the difficulty of providing a systematic classification of anomalies. Despite numerous attempts (Bulusu et al. 2020; Chalapathy et al. 2019; Chandola et al. 2009; Goldstein and Uchida 2016; H. Wang et al. 2019), there is no single commonly accepted classification method. One of the most often cited anomaly classifications was presented by Chandola et al. 2009:

1. Point anomaly, when an individual data instance deviates from other data,
2. Collective anomaly, when a collection of data instances is anomalous with respect to the entire dataset,
3. Contextual anomaly, when a data instance is anomalous only in a specific context but not otherwise.

An illustration of the first scenario is a single fraudulent bank transaction within a sequence of normal operations. This is also the most studied case, with many methods, particularly older ones, focusing on the detection of point anomalies. The malicious behaviour of a device or a server, when a sequence of requests or actions comprises an anomaly, is an example of the second case, a collective anomaly. In the third scenario, a credit card transaction for a large sum that may be expected during a specific period, such as the end of the year, but occurs outside of that period, could be considered a contextual anomaly.

An alternative way of looking at anomaly categories, linked to the previous classification, distinguishes between global and local anomalies (Goldstein and Uchida 2016; Schubert et al. 2014; C. Wang et al. 2018). Global anomalies, meaning the data instances that can be viewed as anomalies when looking at the entire population, are equivalent to point anomalies. Contextual anomalies are regarded as local anomalies, as they exhibit abnormal properties within their local neighbourhoods but not necessarily at a global level (C. Wang et al. 2018). The above characteristics are reflected in the landscape of anomaly detection methods; for example, a popular AD algorithm, Local Outlier Factor (LOF) (Breunig et al. 2000), is designed to focus on detecting local anomalies.

Ruff et al. 2021 introduce a new classification aspect resulting in two additional types of anomalies: low-level sensory anomalies and high-level semantic anomalies (Ahmed et al. 2020a). This categorisation reflects high and low-level features, and it is therefore relevant in the context of deep learning-based AD methods. The low-level features refer to pixel-level anomalies in image recognition, whereas the high-level ones refer to whole anomalous objects. Low-level and high-level anomalies in the linguistic environment can be expressed by typographic errors and bigger anomalous language constructs, respectively.

Other authors identify factors, such as the intention of anomaly inclusion, besides the classification types mentioned above. The following are examples of abnormalities in this regard:

- intentional anomalies – introduced by an adversarial process, such as network intrusions; they are also called adversarial examples,
- unintentional anomalies – often called novelty or out-of-distribution instances (Bulusu et al. 2020; Ruff et al. 2021).

Table 2.1 illustrates the variety of anomaly classification aspects found in the literature on anomaly detection. Identifying anomalies in all their manifestations places high demands on algorithms that are expected to address the complexity of the AD problem.

Table 2.1: Anomaly classification aspects existing in the literature.

Classification aspect	Anomaly types	
Spatial scope	Global, point	Local, contextual
Cardinality	Single	Collective
Data complexity	Low-level, sensory	High-level, semantic
Intentionality	Intentional	Unintentional

2.1.3 Anomaly Detection Algorithms

To date, a plethora of algorithms have been proposed to tackle the challenge of anomaly detection. These algorithms vary significantly in complexity, ranging from basic statistical methods to sophisticated deep neural network-based architectures. This section provides an overview of several popular anomaly detection algorithms. The meta-learning experiments conducted in this study utilised all the algorithms discussed here, along with a few others.

k-Nearest Neighbours / k$^{\text{th}}$-Nearest Neighbour (kNN/k$^{\text{th}}$NN)

kNN and k$^{\text{th}}$NN introduced in 2000 (Ramaswamy et al. 2000) belong to a popular nearest neighbours-based family of methods. They require the calculation of distances between all data instances, which makes them computationally intensive for large datasets. There are two common variants of this method – either the distance between the point and its k^{th} nearest neighbour is calculated (k$^{\text{th}}$NN) or an average distance between the point of interest and all its k nearest neighbours (kNN). The distance is used as an anomaly score. Similar to many others, this method requires the parameter k, which must be given to the algorithm in advance.

Local Outlier Factor (LOF)

This density-based method, introduced in 2000 (Breunig et al. 2000), is one of the most popular outlier detection methods, though it mostly focuses on finding local anomalies. It considers k nearest neighbours of an instance and calculates local densities for all of them. The LOF is then obtained as an average ratio of local densities.

One-Class Support Vector Machines (OC-SVM)

This classification-based method can be seen as one of the best-performing methods in many domains (Emmott et al. 2015; Goldstein and Uchida 2016). The method called SVM originating from Vapnik's theory (Vapnik 2013) has been adopted for discovering outliers (Schölkopf et al. 2000) to result in One-class SVM. The goal of this method is to establish a possibly small region (e.g. a hypersphere), which encapsulates all normal instances. The instances falling outside of this region are considered outliers, and the distance from the established region can be used as an anomaly score.

Isolation Forest (IForest)

Isolation Forest (F. T. Liu et al. 2008) has been demonstrated to be one of the most effective AD techniques (Emmott et al. 2015). The algorithm isolates data points at random, and the instances that are easier to isolate are considered more likely outliers.

Histogram-based Outlier Score (HBOS)

This method from a statistical/probability-based family has been proven as one of the fastest AD methods, outperforming many other methods by orders of magnitude in terms of time required (Goldstein and Dengel 2012). The anomaly score is assessed from the size of the histogram bin that a given data point falls into; the smaller the bin, the higher the anomaly score. The number of bins into which the data is split has to be specified in advance.

Copula-based Outlier Detection (COPOD)

COPOD is another probability-based method, which has been recently introduced (Z. Li et al. 2020). This method first constructs an empirical copula and then uses it to predict the tail probabilities of each given data point to determine its level of "extremeness". Similarly to HBOS, it is very computationally efficient.

Autoencoder (AE)

Autoencoders are used in many deep learning methods, either alone or as part of a model architecture (Generative Adversarial Networks – GAN, or Variational Autoencoders – VAE) (An et al. 2015; Di Mattia et al. 2019; Han et al. 2021; Kaplan et al. 2020; Sakurada et al. 2014; Zhou et al. 2017). Autoencoders are neural networks that encode the input to a latent space and then attempt to reconstruct the input from the encoded features. Its potential in discovering anomalies lies in the expectation of reproduced anomalous instances differing largely from actual input data compared to normal data. The reconstruction error is used as the anomaly score. A plain autoencoder architecture has been chosen for this study as one of the deep learning-based techniques.

Single-Objective Generative Adversarial Active Learning (SO-GAAL)

The architecture proposed by Y. Liu et al. 2019 utilises a GAN-like adversarial network involving the generator and the discriminator. The creators advocate the method can directly generate informative potential outliers based on the min-max game between adversarial elements of the architecture.

2.1.4 Classification of Anomaly Detection Algorithms

The existence of a diverse array of algorithms has prompted numerous attempts to categorise these methods into broader families, thereby providing a clearer overview of the landscape of these techniques. However, there is a lack of consensus regarding the classification of AD methods, as different authors employ various criteria to organise this vast diversity of techniques. One comprehensive survey (H. Wang et al. 2019) groups AD algorithms into:

- Statistical-based methods, e.g., Gaussian Mixture Model-based (GMM) (Yang et al. 2009), where identification of an anomaly depends on its relationship to the distribution pattern of other data instances.

- Distance-based methods, e.g., k-Nearest Neighbours (kNN) (Ramaswamy et al. 2000), where a data point is classified as an anomaly based on its distance to neighbourhood points.
- Density-based methods, e.g., Local Outlier Factor (LOF) (Breunig et al. 2000) or Connective-based Outlier Factor (COF) (Tang et al. 2002), for which anomalies are identified as instances occurring in low-density regions, as opposed to regular data instances.
- Clustering-based methods, e.g., Cluster-Based Local Outlier Factor (CBLOF) (He et al. 2003), where anomalies are identified as those that occur outside of the determined data clusters.
- Graph-based methods that capture the dependencies of linkages between instances in order to discover those with fewer interconnections.
- Learning-based methods including deep-learning techniques that learn a model to identify anomalies.
- Ensemble-based methods that combine the results of different techniques to identify anomalies.

Aside from the groupings outlined above, further categories have been proposed in studies that have mostly focused on classic methods:

- Nearest neighbour-based (Chandola et al. 2009; Emmott et al. 2015; Goldstein and Uchida 2016), e.g., kNN, Angle-Based Outlier Detection (ABOD) (Kriegel et al. 2008),
- Classifier-based (Chandola et al. 2009; Goldstein and Uchida 2016), also referred to as model-based (Emmott et al. 2015), or learning-based methods (H. Wang et al. 2019),
- Subspace-based or spectral approaches, in which anomalies are easily discovered following dimensionality reduction or identification of a specific low-dimensional subspace (Chandola et al. 2009; Goldstein and Uchida 2016),
- Projection-based, e.g., Isolation Forest (IForest) (F. T. Liu et al. 2008) or Lightweight On-line Detector of Anomalies (LODA) (Pevnỳ 2016) – methods using information

Table 2.2: Examples of AD methods arranged according to two-dimensional classification, proposed by Ruff et al. 2021. Acronym dictionary: OC-SVM—One-class support vector machine, SVDD—Support vector data description, SSAD—Semisupervised AD, GMM—Gaussian mixture model, PCA—Principal component analysis, rPCA—Robust PCA, k-NN—k-nearest neighbours, LOF—Local outlier factor, IForest—Isolation forest, OC-NN—One-class neural network, DSVDD—Deep support vector data description, GT—Geometric transformation, EBM—Energy-based model, AAE/CAE—Adversarial/Contrastive autoencoder, GAN—Generative adversarial network.

		Model			
		Classification	Probabilistic	Reconstruction	Distance
Feature map	Shallow	OC-SVM	Histogram	PCA	k-NN
		SVDD	Mahalanobis	rPCA	LOF
		SSAD	GMM	k-Means	IForest
	Deep	OC-NN	EBMs	AAEs	
		DSVDD	Flows	CAEs	
		GT		GAN	

from random projections of the data (Emmott et al. 2015),

- Information Theoretic techniques – methods that make use of information theoretic measures, such as *entropy* or *Kolomogorov Complexity* metric (Chandola et al. 2009).

An alternative approach to categorising AD methods has been proposed by Ruff et al. 2021. These researchers suggest a two-dimensional classification framework based on the model and the feature map, as shown in Table 2.2. Distinguishing between shallow and deep feature maps while retaining the classification based on model characteristics makes this categorisation particularly successful in communicating the multifaceted nature of the problem.

2.1.5 Benchmark Datasets

The distinction between a "real-life dataset" and a "synthetic dataset" that is commonly used in literature is insufficient for the AD context. For AD purposes, many researchers focus on a more nuanced differentiation that better suits AD challenges. This involves distinguishing between datasets repurposed from classification tasks and "semantically meaningful" datasets (Campos et al. 2016; Ruff et al. 2021; H. Wang et al. 2019).

In cases of data repurposing, even though the dataset might originate from real data, the classes designated to represent anomalies do not necessarily reflect real-life anomalies; they are merely different classes. The process of dataset creation is typically structured in a way that the classes intended to represent anomalies differ markedly, and their instances occur less frequently compared to “normal” class instances. However, such data may not fulfill the criterion of diversity (Campos et al. 2016). Additionally, it may fail to exhibit other characteristics of anomalies, such as those arising from anomalies' unpredictable nature. An example of a repurposed dataset could be one containing handwritten letters in which instances of selected characters are down-sampled to form a minority class. Although some researchers have criticised this practice (Campos et al. 2016), it remains one of the most widely used techniques for generating benchmark datasets for evaluating new AD algorithms (Goldstein and Uchida 2016; Kandanaarachchi et al. 2020). The methodologies of creating AD benchmark datasets have been studied and proposed by Emmott (Emmott et al. 2013, 2015).

In semantically meaningful datasets, the instances of minority class are expected to happen rarely in real-life situations, for example, people with specific medical issues among healthy individuals or fraud transactions among routine bank transactions (Campos et al. 2016).

Several AD benchmark datasets, including repurposed and semantically meaningful datasets can be obtained from public repositories: Outlier Detection Data Sets (Campos et al. 2016), Outlier Detection DataSets (ODDS) (Rayana 2016), Unsupervised Anomaly Detection Benchmark (Goldstein 2015; Goldstein and Uchida 2016), Anomaly Detection Meta-Analysis Benchmarks (Emmott et al. 2015).

2.1.6 Evaluation Metrics

The output of an AD method is usually given by an anomaly score, which indicates how likely it is that an observation is anomalous (Campos et al. 2016; Goldstein and Uchida 2016). Generally, the scores generated by various methods are not normalised; they can take any real number as their value, including negative ones. As a result, they cannot be easily read as probabilities or directly compared across methods. To facilitate comparison of individual results, ranking techniques are essential. To assist with the comparison of performance of several algorithms on a dataset, additional metrics are required.

When assessing an algorithm's performance on datasets with known ground-truth labels, common evaluation metrics for classification problems – such as precision, recall, and F_1 score – are also employed in AD tasks. Nonetheless, these metrics require the predicted scores to be translated into binary labels. This translation necessitates the use of a threshold value that is task-specific. A popular technique for estimating this threshold is the utilization of the number of ground-truth outliers, n, which enables the computation of these metrics.

Another metric often utilised in AD scenarios is Precision at n (P@n), which is measured as the percentage of correctly identified outliers among the top n ranks (Campos et al. 2016).

Weighted correlation coefficient, such as Spearman's rank similarity or Pearson correlation, is another performance metric utilised in AD research (H. Wang et al. 2019). These metrics leverage the correlation between predicted anomaly scores and the ordered sequence of actual outlier and inlier instances. In the calculation of this metric, greater weight is given to instances assigned higher outlier scores. Consequently, the method imposes a more severe penalty for errors made on observations that were attributed the highest scores.

The Area under the Receiver Operating Characteristic (ROC) curve (AUC) and the Average Precision (AP) are widely used evaluation measures in anomaly detection since they provide the integral result for all thresholds from 0 to 1, making them threshold-independent.

A ROC curve is obtained by plotting the True Positive Rate (TPR) versus False Positive Rate (FPR) for the full range of thresholds. TPR and FPR are defined as $\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}}$, $\text{FPR} = \frac{\text{FP}}{\text{FP}+\text{TN}}$, where TP – true positives, FP – false positives, TN – true negatives, and FN – false negatives. The area under the ROC curve serves as a single performance metric.

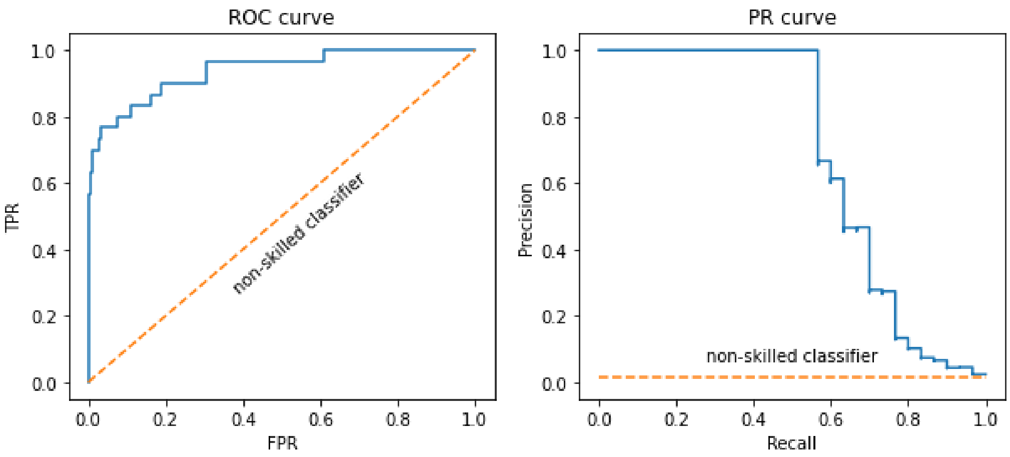


Figure 2.1: ROC and Precision-Recall (PR) curves.

For a non-skilled classifier, the value of AUC is expected to be about 0.5; for the ideal classifier, it is a value of 1.

A Precision-Recall curve is created by plotting precision versus recall for the entire range of thresholds. These two measures are defined as $\textsc{Precision} = \frac{\text{TP}}{\text{TP+FP}}$, $\textsc{Recall} = \text{TPR}$. The area under this curve is known as the Average Precision (AP) and can be expressed as:

$$\text{AP} = \sum_k P_k \Delta R_k \tag{2.2}$$

where P_k is the precision at the k^{th} threshold and ΔR_k is the change in recall between thresholds $k-1$ and k. While for the ideal classifier the AP value equals 1, the non-skilled classifier reports a value close to the proportion of the anomalies within the dataset. The examples of a ROC curve and a Precision-Recall curve are presented in Figure 2.1.

These two metrics are commonly used in AD research because they effectively gauge the performance of individual AD algorithms. Unlike metrics that rely solely on fixed binary labels, AUC and AP consider the entire range of possible thresholds, providing a more comprehensive assessment of the predictive capabilities of an algorithm.

In summary, this section has offered a thorough investigation of anomaly detection, shedding light on the aspects required for presenting the primary goal of the research. The provision of such a rich context will aid in the subsequent presentation of the work, of which the AD is a foundational pillar.

2.2 AutoML in Supervised Learning

To address the Algorithm Selection Problem in AD, it is critical to explore existing knowledge on AutoML and meta-learning, which serve as the second foundation of this research. This section, and the following (Section 2.3), provide the overview of the AutoML research in both supervised and unsupervised aspects, respectively.

Since the formalisation of the ASP framework by Rice (Rice 1976), meta-learning has received increased attention in the research community. Studies have focused on specific areas of meta-learning, such as hyperparameter optimisation (HO) (Feurer, Springenberg, et al. 2015; Horváth et al. 2016; Komer et al. 2014; Rafael G Mantovani et al. 2015, 2019; Rafael Gomes Mantovani et al. 2020; Sanders et al. 2017; Wistuba et al. 2016) or meta-features development (Abdrashitova et al. 2018; Gutierrez-Rodríguez et al. 2019; Kanda et al. 2016). Several survey studies have presented a large-scale overview of the research (I. Khan et al. 2020; Lemke et al. 2015; Muñoz et al. 2015; Smith-Miles 2009; Vilalta et al. 2002). In particular, a comprehensive compilation of the research and accomplishments in AutoML with a focus on meta-learning is presented by Hutter et al. 2019. Current AutoML systems are not only capable of performing model or hyperparameter selection tasks, but are also fully functional automated pipelines of processes that include training and testing without the need for human intervention (Guyon et al. 2019). To date, several automated AutoML systems have been developed, including Auto-WEKA (Kotthoff et al. 2019), Auto-Sklearn (Feurer, Klein, et al. 2015), Auto-Sklearn 2.0 (Feurer, Eggensperger, et al. 2022), Hyperopt-Sklearn (Komer et al. 2014), Auto-Net (Mendoza et al. 2019) and others (Olson et al. 2016; Steinruecken et al. 2019).

These systems were designed to address supervised problems, but the ASP for unsupervised settings remains largely unexplored. The approaches common to supervised AutoML, such as Bayesian optimisation, population-based search used in evolutionary techniques, grid or random search (Bahri et al. 2022; Hutter et al. 2019), cannot be easily utilised in unsupervised scenarios due to the lack of a simplistic performance metric.

2.3 AutoML in Unsupervised Anomaly Detection

To comprehensively present the current research on automated approaches to unsupervised AD, this section reviews the literature on AutoML in AD, not necessarily unsupervised, and also in other unsupervised scenarios, not necessarily related exclusively to AD.

A comprehensive end-to-end system similar to those mentioned in the previous section has been proposed by Y. Li et al. 2020. It provides an automated pipeline for anomaly detection, including algorithm search, hyperparameter search, and data visualisation. It has support for stationary and time-series data. However, even though the algorithm search space is built upon unsupervised AD algorithms, the algorithm and hyperparameter selection is performed using the F_1 score metric, which requires the ground truth labels for its calculation. As a result, it more closely resembles the methodology used in supervised systems rather than employing techniques tailored for unsupervised scenarios.

The subsequent review focuses on the approaches intended to address the unsupervised aspect of AD problems. It is conducted by reviewing the meta-learning framework adapted by individual studies, followed by the review of meta-characteristics, which constitute one of the essential components of a meta-learner framework.

2.3.1 Meta-Learning Frameworks for Algorithm Selection in AD

According to I. Khan et al. 2020, the important dimensions of the meta-learning framework are: meta-features, meta-model and a meta-target, where for any given problem, the meta-learner receives meta-features as input and recommends appropriate algorithms according to the learned meta-model. However, the approaches to algorithm selection for AD do not always include all the elements of the indicated framework.

The approaches proposed by Ferdosi et al. 2019 and Gudur et al. 2022 do not offer any particular meta-model. Ferdosi et al. 2019 performs the algorithm search over the labelled training dataset to find the best performing AD algorithm. This algorithm is then applied to the unlabelled test dataset, and the outlier scores are obtained. The outlier visualisation and the feedback from a human expert is utilised as a deciding factor whether a given algorithm performed well, and to refine the result of the proposed algorithm, which in turn obtains better outlier ranking. Similarly, Gudur et al. 2022 use the feedback from a human expert obtained on a subset of datasets. That subset is chosen using the pre-designed acquisition function and then employed in a grid search-based technique to determine the optimal algorithms. Both studies are primarily concerned with reducing the amount of data supplied to the human expert for labelling. However, they keep humans in the loop, which is a significant constraint for process automation.

Meta-AAD (Zha et al. 2020) leverages reinforcement learning to develop a meta-policy using labelled AD data. The meta-policy is then applied to a new, unlabeled dataset, and human feedback is used to confirm the expected anomalies and to further refine the meta-policy.

A different approach to addressing the ASP in AD to that mentioned above is proposed by Papastefanopoulos et al. 2021 and Le Clei et al. 2022. These works propose automated approaches while taking into consideration the unsupervised nature of AD problems, but they do not perform offline meta-training. Instead, the proposed systems assess the performance of a predefined set of algorithms “online”, using “voting” procedures, during an algorithm evaluation over a new task. The primary shortcoming of these systems is non-scalability. They require evaluation (or even multiple evaluations) of a number of algorithms when confronted with a new problem without taking advantage of transfer learning from historically evaluated algorithms.

Kandanaarachchi et al. 2020 proposed an SVM-based meta-model trained on dataset meta-features, with the target variable being a binary label of good (AUC >= 0.8) and bad algorithm performance. They feed two meta-features into the meta-model. These features were subsequently used to generate a two-dimensional map identifying the optimal regions for each algorithm. However, the final output from the SVM does not correlate well with the original performance data. This suggests that the binary classifier based on two-feature input is overly simplistic in capturing the complicated relationship between dataset attributes and algorithm performance.

Table 2.3: Overview of the approaches proposed by MetaOD and FMMS.

	MetaOD	FMMS
Meta-features	200 features	46 features
Meta-model	Matrix Factorisation	Factorisation Machine
Meta-target	AP	*Rank* and *TopNK*
Dataset count	162	553
Model count in a base set	302	200

The study by Zhao, Rossi, et al. 2021 (MetaOD) and the Factorisation Machine-based Model Selection (FMMS) approach (Zhang et al. 2022) both utilise the framework outlined by I. Khan et al. 2020, incorporating a set of meta-features, a meta-model, and a meta-target. In terms of meta-features, both studies employ statistical and "landmarking" features. Statistical features, such as minimum, maximum, or variance, are used to characterise the underlying data distributions. The landmarking features are designed to reveal the internal structure of the datasets (Zhang et al. 2022). Both MetaOD and FMMS adhere to the underlying assumption that the algorithm selection problem is conceptually similar to recommender systems and collaborative filtering. MetaOD's meta-model is based on matrix factorisation, whereas FMMS employs a factorisation machine. The metrics used by these approaches include AP (as detailed in Equation 2.2), *Rank*, and *TopNK*. The *Rank* metric is quantified as a subsequent value for each recommended algorithm, expressed as a fraction of the total number of algorithms. *TopNK* represents the likelihood that a model ranked within the top N algorithms will be predicted among the top K algorithms. The overview of both techniques are presented in Table 2.3.

Both studies are designed for unsupervised scenarios and represent state-of-the-art solutions in this field. Nonetheless, they exhibit certain limitations. A primary limitation is the relatively limited dataset collections, particularly in light of the meta-analytic character of this research. Another shortcoming pertains to the choice of evaluation metrics. Notably, neither study employs the AUC metric, which is among the most prevalent metrics in AD research (Campos et al. 2016; Goldstein and Uchida 2016; H. Wang et al. 2019). The ranking-related metrics used in the FMMS study additionally suffer from a constraint, offering limited insight into the actual distribution of various algorithms' performance. For instance, the third-ranked algorithm could exhibit performance closely similar to the first-ranked one, yet this nuance is not visible through ranking metrics alone.

2.3.2 Meta-Characteristics of Datasets

The ability to characterise datasets using a common set of meta-features is one of the key components of all automated algorithm selection methods. A key early exploration of this challenge by Campos et al. 2016 examined how well the selected AD methods handled anomalies across datasets. The authors defined two dataset properties – *difficulty* (Diff) and *diversity* (Diver). For a dataset x and a set of AD algorithms $\alpha_j \in \{\alpha_1, ..., \alpha_L\}$, with the algorithm performance y and the standard deviation of performance across algorithms $\sigma_j(y)$, the characteristics are described by Equations 2.3 and 2.4.

$$\textsc{Diff}(x) = 1 - \frac{1}{L}\sum_{j}^{L} y(x, \alpha_j) \tag{2.3}$$

$$\textsc{Diver}(x) = \sigma_j\big(y(x, \alpha_j)\big) \tag{2.4}$$

The difficulty score indicates the degree of difficulty for an AD algorithm to detect outliers in a given dataset. The diversity score reflects the agreement between the algorithms on the difficulty score. The researchers created a feature map based on these two scores and positioned datasets accordingly. As these characteristics were derived from algorithm evaluations, they would not qualify for use in a meta-learner; however, this meta-analysis of the algorithm's performance across datasets is seen as a first step towards inferring dataset meta-features and developing a system to aid in algorithm selection.

Many common attempts to construct dataset meta-features specifically for anomaly detection tasks rely on labelled data nonetheless. Kandanaarachchi et al. 2020 created the feature set with 362 features, which was subsequently reduced to two features and adapted to a two-dimensional feature map. Because labelled data is required for a significant part of the initial set, it is inadequate for an unsupervised AD problem. Kotlar et al. 2021 proposed an alternative set of meta-features. Their study centred on semantic features specifically developed for AD, such as global, local, or collective anomaly types, as well as other characteristics, such as anomaly space, anomaly ratio, data type, or data domain. To create these features, prior knowledge of anomalous instances, such as their ratio or distribution, is required. These characteristics must also be provided by a human expert rather than being retrieved automatically from datasets. As a result, this technique cannot be immediately applied to an automated unsupervised meta-learning problem.

Zhang et al. 2022 adopted the set of meta-features initially proposed by Fusi et al. 2018, which are commonly used in classification tasks. These features are label-independent, making them well-suited for unsupervised model selection techniques. Label-independent meta-features were also proposed by Zhao, Rossi, et al. 2021 in their unsupervised AD algorithm selection system. In addition to the statistical meta-features commonly used in classification tasks, this feature set also incorporates "landmarking" features that are specifically tailored for AD problems. The proposed set contains 200 meta-features in total.

As mentioned in Section 2.3.1, one of the main limitations among the reviewed studies is the narrow scope of datasets employed in their experimental work. While most of these studies relied on a very limited set of datasets, the most extensive studies included either 553 (Zhang et al. 2022) or 162 (Zhao, Rossi, et al. 2021) datasets. Notably, one study that published a collection of 12,000 datasets (Kandanaarachchi et al. 2020) failed to fully leverage its potential by severely limiting the feature space and oversimplifying the results from a binary classifier.

In summary, a persistent challenge in the field of AD is the lack of comprehensive research on automated algorithm selection. To date, the most promising approaches for addressing this challenge are the MetaOD (Zhao, Rossi, et al. 2021), and the conceptually comparable FMMS (Zhang et al. 2022) framework. In the current study, the MetaOD is investigated further and used as a baseline as the first attempt to algorithm selection in truly unsupervised anomaly detection.

2.4 Evaluation of Local Reliability

Traditional metrics used in evaluating the efficacy of machine learning models often focus on the model's global performance. This is commonly done by averaging errors across all data points, exemplified by metrics such as mean squared error (MSE) or mean absolute error (MAE), or by calculating the overall accuracy over an entire dataset. Contrary to assessing global performance of models, the term "reliability" is used by some researchers to refer to the quality of individual predictions, or localised performance (Bosnić et al. 2009). Measuring localised performance is particularly relevant for meta-learners, where the main focus is on determining the reliability and trustworthiness of specific responses provided by the meta-learner.

Research into techniques for assessing the local reliability of predictions remains relatively unexplored, particularly in the realm of meta-learning. Several studies have investigated and compared methods for determining local reliability in traditional machine learning algorithms, including linear regression, regression trees, random forests, or SVM (Bosnić et al. 2008a,b, 2009, 2010). Preliminary findings from these studies suggest that variance-based techniques exhibit strong potential in evaluating prediction reliability (Bosnić et al. 2008a).

An alternative methodology presented by Prudêncio et al. 2022 attempted to model the quality of responses using decision trees. This approach utilized two-feature subspaces as input and categorized the responses into 'good' or 'bad' classes. While the simplicity of this technique aligns with certain requirements, especially in data presentation, solely focusing on feature pairs may ignore the complexity inherent in many problems.

Research studies that employ selective classification or selective regression approaches are valuable resources for investigating the reliability of particular responses (Fisch et al. 2022; Jain et al. 2022; Park et al. 2023; Shah et al. 2022; Wiener et al. 2012; Zaoui et al. 2020). Although the majority of this literature addresses classification problems, there are notable studies that apply relevant techniques to regression models (Shah et al. 2022; Zaoui et al. 2020). These studies used the *conditional variance function* as a measure of uncertainty of individual predictions (Zaoui et al. 2020). They proposed to use the local sample variance of the model's residuals as an estimator for the conditional variance function and utilise this as a measure of uncertainty for particular data instance:

$$\sigma^2(x) = \mathbb{E}\left[(Y - f^*(X))^2 \,|X = x\right] \tag{2.5}$$

with X being a feature vector and Y its corresponding output, and $f^*(x)$ the modelled regression function. Given that the meta-model employed in the current work is a supervised regression model, these investigations are considered relevant.

2.5 Summary

In a review of the existing literature relevant to the subject matter, several notable gaps and opportunities for further investigation can be observed. The problem of algorithm selection for unsupervised AD remains largely underexplored. While a handful of methodologies have been proposed, there is an absence of research focused on identifying the most influential and promising components of a meta-learner with respect to its performance. Additionally, the domain of meta-learning lacks studies aimed at estimating the reliability of task-specific predictions, as well as evaluating the associated risks within the feature space where such tasks are situated. Finally, studies concerning meta-learning for unsupervised problems have been constrained to a narrow range of datasets. All the above findings highlight the importance of delving deeper into this field and developing adequate methodologies.

Chapter 3

Datasets, Metrics, and Preliminary Experiments

Understanding and evaluating the challenges and the potential of AutoML for optimal selection of algorithms for anomaly detection requires both benchmark datasets and a suitable experimental framework. The AD benchmark datasets utilised in this study are presented and evaluated in this chapter and the metrics used to evaluate the performance of AD algorithms are discussed. Central to this research are the preliminary experiments that informed the direction of subsequent investigations. Insights from these tests provided a basis for the hypotheses explored in this research and facilitated the generation of data that was used for the meta-learners' training. These preliminary experiments are described in the final section of this chapter.

3.1 Datasets

The meta-learning experiments carried out in this study involved a few collections of AD benchmark datasets. The term *dataset* is used throughout this thesis to refer to data specific to a particular AD task. To refer to the entire dataset collection, either the term *set* or *collection* is used.

Current research employs the following sets:

- *Kandanaarachchi* set (Kandanaarachchi et al. 2020)

- *Goldstein* set (Goldstein and Uchida 2016)
- IoT botnet attacks (*N-BaIoT*) (Meidan et al. 2018)

Kandanaarachchi is the largest set created to date, containing over 12,000 datasets (Kandanaarachchi et al. 2020). It has been combined from public repositories and repurposed from existing classification datasets by down-sampling certain categories. In this study, this set is used in the main meta-learner experiments.

Similarly to the *Kandanaarachchi* set, the *Goldstein* set has been created mainly by re-purposing classification datasets using a variety of data formats, including images, speech recordings, and numerical data. Several datasets from this collection have been used in the current research for illustrative purposes.

Unlike the two sets above, the *N-BaIoT* data was generated specifically for anomaly detection purposes (Meidan et al. 2018). Because the nature of the data resembles a potential real-life scenario (of a hacker's attack), the data with such characteristics is referred to as *semantically meaningful* data in the context of anomaly detection (Campos et al. 2016). In the current work, this dataset has also been used for auxiliary experiments and illustrative purposes.

Subsequent sections provide a more comprehensive description of the chosen sets, including the motivation for selecting these datasets. Specifically, the *Kandanaarachchi* set is discussed in greater detail, considering various aspects. This in-depth analysis is crucial, as this set forms the basis of the main meta-learning experiments. Understanding its detailed characteristics is essential to justify its application in such research.

3.1.1 Kandanaarachchi Benchmark Set

Over 12,000 datasets make up the largest AD benchmark dataset yet released (Kandanaarachchi et al. 2020). The main motivation for selecting this set for the experiments was its size. As highlighted earlier, all of the experiments within the meta-learning studies used significantly smaller numbers of datasets, which was insufficient for making any statistical observations on the performance. The *Kandanaarachchi* collection was created with the intention of approaching the ASP for unsupervised tasks, however, the

Table 3.1: Statistics of the *Kandanaarachchi* set of datasets.

Statistic	Observations	Features	Anomalies %
Min	44	2	1.34
Median	622	16	4.21
Max	10,460	1,556	5.33

creators have not yet utilised this collection to its full potential for AD and, in particular, AutoML applications.

In addition to the size of the set, an important consideration is the diversity of the datasets it contains, including characteristics such as coverage and variation according to meta-features. Table 3.1 presents the statistics of the datasets used in *Kandanaarachchi* according to the number of observations, number of features and percentage of anomalies contained in the datasets.

The creators of the datasets examined them using characteristics introduced by Campos et al. 2016, such as *difficulty* (Diff) and *diversity* (Diver), as described by Equations 2.3 and 2.4 in Chapter 2:

$$\textsc{Diff}(x) = 1 - \frac{1}{L}\sum_{j}^{L} y(x, \alpha_j)$$

$$\textsc{Diver}(x) = \sigma_j\big(y(x, \alpha_j)\big)$$

where x is a single dataset, and the characteristics are calculated across a set of AD algorithms $\alpha_j \in \{\alpha_1, ..., \alpha_L\}$, with y being an algorithm performance and $\sigma_j(y)$ its standard deviation across algorithms.

These characteristics are the algorithms' responses on each of the datasets. The difficulty score describes how difficult it is for a given AD method to identify the outliers in a given dataset. Higher scores indicate a better blend of outliers and inliers, and thus higher difficulty. Diversity describes the agreement on the difficulty score amongst the AD algorithms. The details of AD algorithms used for obtaining these characteristics are included in the publication by Kandanaarachchi et al. 2020, and the results are shown in Figure 3.1. The individual colours are related to the original source of the datasets.

Anomalies from the datasets in the bottom-left quadrant of the figure are straightforward to recognise for the majority of algorithms since the difficulty and diversity scores are low. The spread of diversity increases considerably as difficulty increases, indicating that while many datasets are relatively easy to interpret for all algorithms, there is also an abundance of datasets on which different algorithms exhibit diverse performance. Lastly, the datasets in the bottom-right corner of the picture are tough for the majority of algorithms. Overall, two key observations emerge from this analysis: first, that the *Kandanaarachchi* set includes datasets that span a wide range of difficulty levels; and second, that there exists a substantial number of datasets where the performance of various algorithms differs significantly.

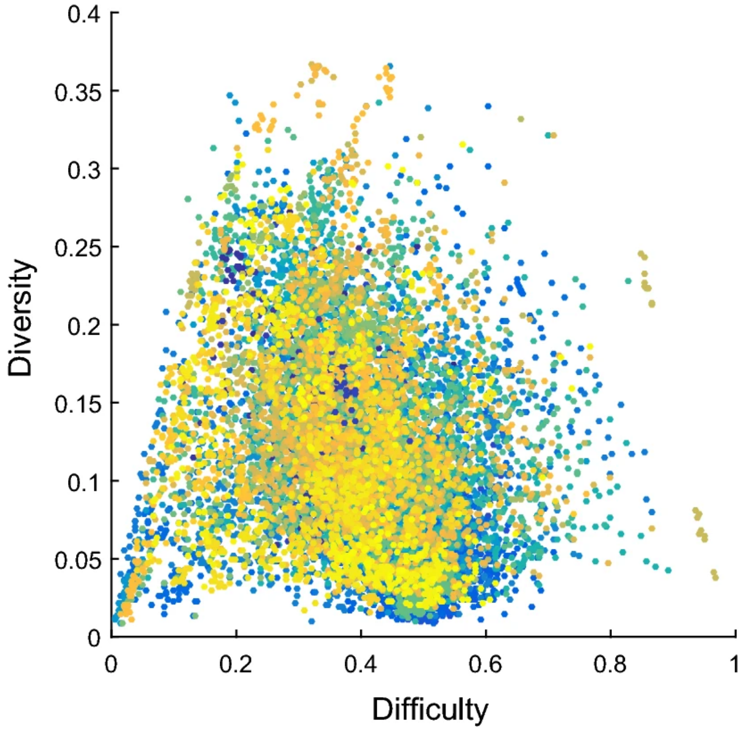


Figure 3.1: Diversity and difficulty as defined by Campos et al. 2016, visualised for the *Kandanaarachchi* set. The image from (Kandanaarachchi et al. 2020).

In addition to the dataset inspection in prior works, this study looked at additional aspects describing the collection's heterogeneity. To achieve this, the dataset collection was arranged inside the meta-feature space constructed from features proposed in the meta-learner experiment presented in this research. The specific meta-features used here are further described in Section 4.1.1.

To illustrate the concept of the population of meta-feature space, the term *coverage* is introduced. A dataset is represented as a single point in a k-dimensional space, with k denoting the number of meta-features. It is assumed that each dataset point encompasses a finite, small area. In this context, the focus is on the space each point covers. When these spaces are projected onto individual meta-features, *coverage* can be defined as the collective area encompassed by all dataset points along each feature.

Figure 3.2 illustrates the *coverage* of the datasets across each meta-feature normalised to the [0, 1] range. Each small bar along a given feature represents a single dataset. Visual inspection suggests that most of the features have a decent representation across the value ranges with the involved datasets.

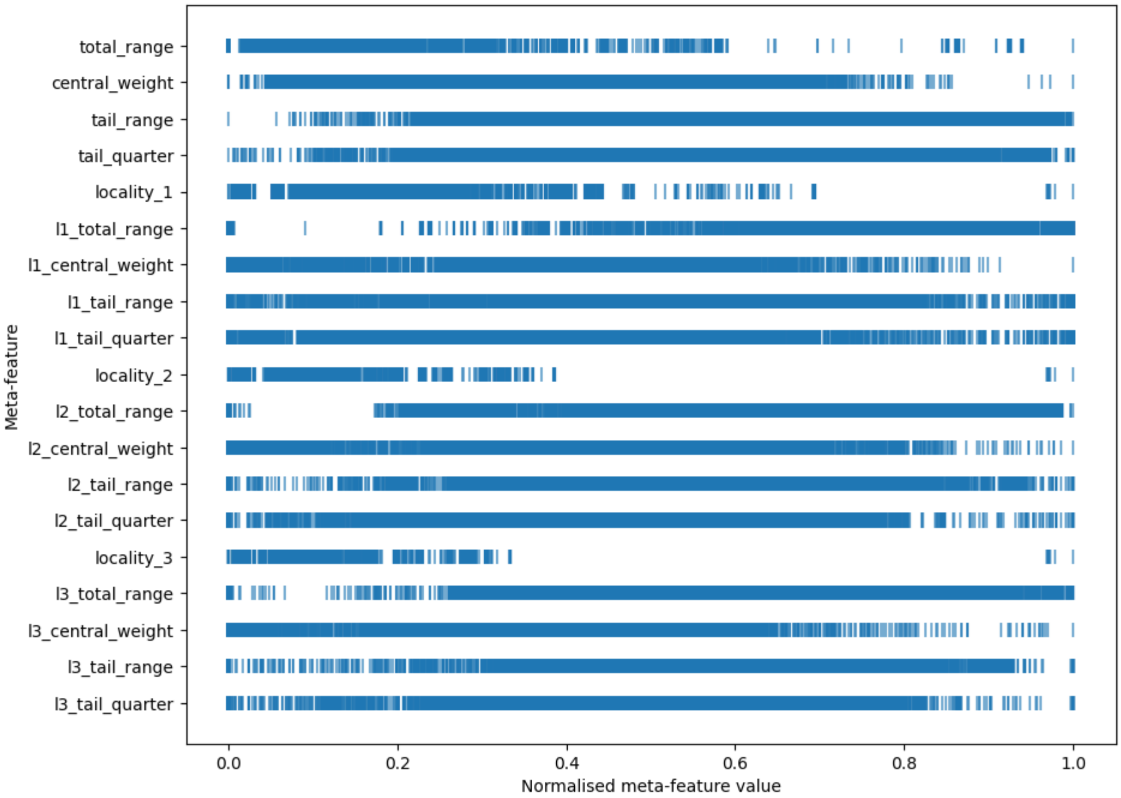


Figure 3.2: Visualisation of dataset *coverage* in the meta-feature space of normalised features.

Apart from visual inspection of Figure 3.2, the *evenness measure* obtained from the Shannon entropy measure (Shannon 1948) was used to quantify the sparsity of the meta-feature space, following the methodology used by Pham et al. 2010 and Bahrpeyma et al. 2021. Equation 3.1 expresses the entropy of a discrete random variable X, where

$X_1, ..., X_S$ denotes the set of possible states of X and $p(X_i)$ represents the probability that $X = X_i$. The entropy of the variable increases with the evenness of its distribution:

$$H(X) = -\sum_{i=1}^{S} p(X_i) \log p(X_i) \tag{3.1}$$

To apply Shannon entropy, which relies on discrete variables, the meta-features were separated into 5 even buckets. Consequently, in this case, $X_i \in \{X_1, ..., X_5\}$ reflects a meta-feature belonging to the bucket i. Following the work mentioned earlier (Pham et al. 2010), the entropy was further normalised to its maximum value to facilitate easier interpretation. This normalised value is also known as the *evenness measure*.

The maximum entropy of any given feature can be expressed as:

$$H_{\text{max}}(X) = -\sum_{i=1}^{S} \frac{1}{S} \log \frac{1}{S} = \log S \tag{3.2}$$

Consequently, the *evenness measure* of a given feature can be written as:

$$H_E(X) = \frac{H(X)}{H_{\text{max}}(X)} = -\frac{1}{\log S} \sum_{i=1}^{S} p(X_i) \log p(X_i) \tag{3.3}$$

The outcomes of applying the Equation 3.3 on the meta-features are presented in Table 3.2. These metrics reveal how evenly the dataset points are distributed within the meta-feature space when they are projected onto individual meta-features. The greater the value, the more uniform the distribution along a particular meta-feature, and hence the better the dataset population or representation. Out of 19 meta-features, 14 exhibit an evenness measure of more than 0.75, with an overall average of 0.715. This indicates that the meta-feature ranges are adequately represented.

After examining the statistics and metrics of the datasets in *Kandanaarachchi*, it has been concluded that this collection is well-suited for advanced meta-learning analysis and evaluation of potential approaches for AutoML.

Table 3.2: Evenness measure of meta-features.

Meta-feature	H_E	Meta-feature	H_E
total_range	0.375	*l2_total_range*	0.983
central_weight	0.789	*l2_central_weigh*t	0.758
tail_range	0.870	*l2_tail_range*	0.833
tail_quarter	0.896	*l2_tail_quarter*	0.825
locality_1	0.424	*locality_3*	0.068
l1_total_range	0.888	*l3_total_rang*e	0.978
l1_central_weight	0.756	*l3_central_weight*	0.674
l1_tail_range	0.892	*l3_tail_range*	0.829
l1_tail_quarter	0.839	*l3_tail_quarter*	0.825
locality_2	0.082		
		Average	0.715

3.1.2 Goldstein Benchmark Set

The *Goldstein* set, originally derived from publicly available classification datasets, was specially processed for benchmarking unsupervised AD algorithms, as detailed by Goldstein and Uchida 2016. This set comprises 10 diverse datasets, including *Breast Cancer*, *Handwritten Digits Recognition*, *Letter Recognition*, *Speech Accent*, *Statlog (Landsat Satellite)*, *Statlog (Shuttle)*, *Thyroid Disease*, *ALOI (Object Images)*, and *KDD Cup 1999*.

In creating this benchmark collection, the authors downsampled selected classes to form minority classes, as described by Goldstein and Uchida 2016. Several of the resulting datasets contain minority classes that represent real-world anomalies. The examples of such include disease data amidst healthy patient data, or instances of network attacks within regular traffic data.

The *Goldstein* set has established itself as one of a benchmarking standards in AD, widely utilised in numerous research studies (Bauder et al. 2017; Guo et al. 2019; S. S. Khan et al. 2018; H. Wang et al. 2019; Yao et al. 2019). Due to this reason it was leveraged in the current research for benchmarking selected AD algorithms and for illustrative purposes.

3.1.3 N-BaIoT Benchmark Set

The N-BaIoT set has been collected by measuring the traffic sent by commercial IoT devices before and after the deployment of two popular botnets (BASHLITE and Mirai). The details on the botnet deployment, measurement, and data collection are given by Meidan et al. 2018. The key rationale for choosing this set in the experiments was that it is "semantically meaningful," which means that these types of anomalies are possible in real life.

The dataset contains data from nine IoT devices: two doorbells, a thermostat, a baby monitor, four security cameras, and a webcam. The data from each device comprises benign traffic and two types of botnet attacks: BASHLITE and Mirai. Each botnet produced five types of malicious behaviour.

Originally, the published data contained separate files with benign traffic for each device and additional files with two types of traffic from infected devices. The sample datasets for the current study were created in the following way: for each device and each type of traffic, the random samples of benign and malicious data were collected and merged together. The benign traffic sample size varied randomly between 5,000 and 20,000, and the amount of malicious traffic varied from 1% to 10% for each dataset size. This preprocessing allowed for exploiting this set in AD algorithm benchmarking studies while considering this data to be "semantically meaningful", because it reflects a potential real-life scenario of an attack on an existing commercial IoT device.

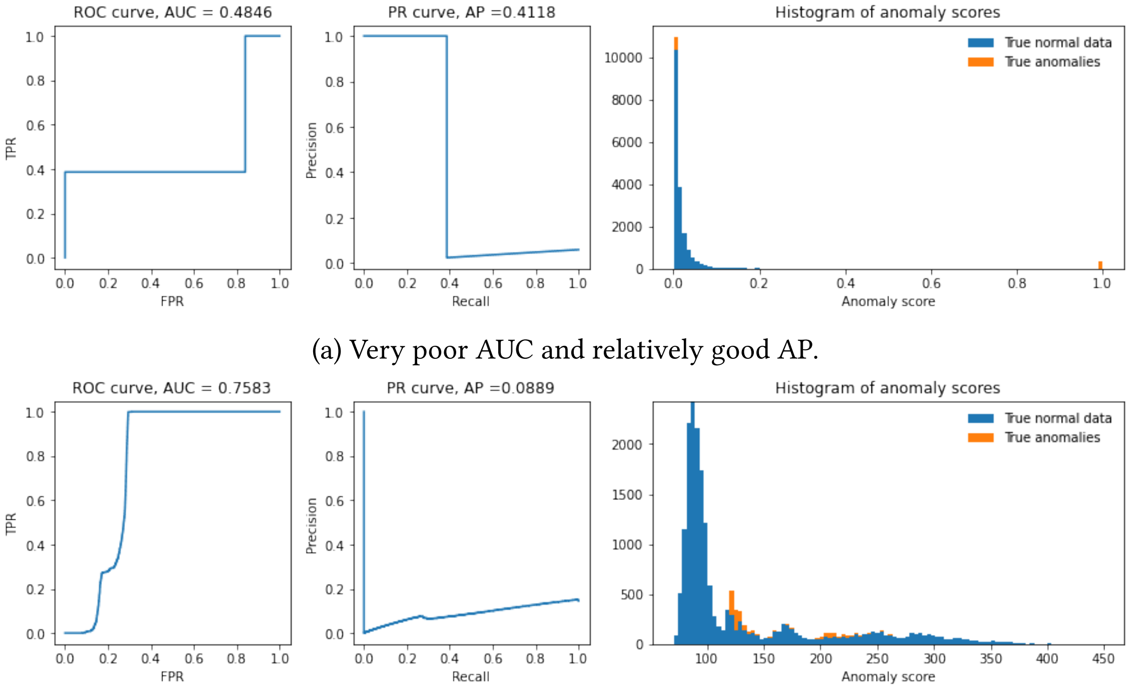
3.2 Anomaly Detection Metrics

Area Under the Receiver operating characteristic (ROC) Curve (AUC) and Average Precision (AP) were chosen to evaluate the performance of AD algorithms in this work since they are the most extensively used evaluation measures in anomaly detection literature (Campos et al. 2016; H. Wang et al. 2019). As mentioned in Section 2.1, the output of an AD method is typically an anomaly score, which indicates the likelihood that the given observation is anomalous. The translation of anomaly scores into binary labels that are required in common performance metrics, such as accuracy or precision, involves using a threshold, which in many practical AD cases is unknown without additional data exploration. The AUC and AP measures provide the integral result for all thresholds from 0 to 1, making them threshold-independent.

Both metrics capture different aspects of an algorithm's ability to separate normal and anomalous data points; therefore, in many cases is not sufficient to interpret the performance results with the use of a single metric only. Figure 3.3 depicts two examples of datasets in which these metrics appear to contradict one another. The figure presents the ROC and Recall-Precision curves together with a histogram of true anomalies and "normal" instances distributed along predicted anomaly scores. The desired outcome is the greatest divergence between the anomaly scores of anomalous and the normal data.

Figure 3.3a presents a very poor AUC value (less than 0.5), however, the moderate AP indicates that the algorithm performed effectively to some extent. This assumption is confirmed by the profile of anomaly scores, showing that roughly half of the anomalies are separated very well, whereas the other half is blended with normal data. In contrast, Figure 3.3b presents results where the AUC is reasonably good (higher than 0.75), but the AP value is very poor. The histogram of the scores reveals that the fairly high AUC value is driven by a large number of true negatives rather than the well-performing algorithm. The algorithm performs poorly by assigning the range of the highest anomaly scores (250-450) to normal instances, whereas anomalous data are blended with normal instances.

A commonly highlighted disadvantage of the AUC is that it can produce overly optimistic results for severely unbalanced classes due to the dominance of true negatives (Ahmed et al. 2020b; Davis et al. 2006; Ruff et al. 2021), as seen in Figure 3.3b. The AP, in contrast to the AUC, only examines the positive class, hence it provides more information regarding performance over unbalanced classes. The direct comparison of the AP measures across various datasets is limited, however, because the lower bound is not normalised and is tied to the ratio of anomalies within the dataset. Both metrics have been chosen in this research to compensate for their respective limitations.



(a) Very poor AUC and relatively good AP.

(b) Relatively high AUC and very poor AP.

Figure 3.3: Characteristics of AP and ROC AUC.

3.3 Performance of Anomaly Detection Algorithms

In addition to the complexity of measuring and comparing the performance of anomaly detection algorithms, another challenge is that a single algorithm may perform well on one dataset but poorly on another. As previously stated (Section 1.2), the research community agrees that no single algorithm outperforms others on all AD tasks (Campos et al. 2016; Emmott et al. 2015; Goldstein and Uchida 2016; Kandanaarachchi et al. 2020; Zhao, Rossi, et al. 2021). Furthermore, the “difficulty-diversity” plot (Figure 3.1) demonstrates that the performance of many algorithms varies significantly across datasets. This is also supported by the current study’s experiments. Tables 3.3 and 3.4 contain the results of algorithm benchmarking against sample datasets. The datasets for this illustration have been sourced from the *Goldstein* set (Goldstein and Uchida 2016) and the *N-BaIoT* set (Meidan et al. 2018). The algorithms used in this experiment are described in Section 2.1.3.

The striking observation from data in Tables 3.3 and 3.4 is that there is no clear winner among the tested algorithms. Several methods achieve very good separation of anoma-

Table 3.3: Results from the comparative analysis: AUC. Highlighted in bold are the top two best values for each dataset. The upper section contains datasets from the *Goldstein* set, the lower one from the *N-BaIoT* set.

Dataset	IForest	OCSVM	kNN	kthNN	LOF	HBOS	COPOD	AE	Hybrid	SOGAAL
breast-cancer	0.988	**0.991**	0.980	0.983	0.984	0.975	**0.990**	0.981	0.858	0.000
pen-global	0.922	**0.996**	**0.978**	0.900	0.817	0.676	0.785	0.903	0.623	0.324
letter	0.612	0.528	**0.835**	0.777	**0.831**	0.571	0.560	0.572	0.479	0.347
aloi	0.538	0.546	**0.619**	0.589	**0.726**	0.493	0.514	0.556	0.494	0.535
pen-local	0.746	0.606	**0.973**	0.953	**0.976**	0.718	0.525	0.641	0.437	0.162
annthyroid	0.639	0.583	0.672	0.645	**0.738**	**0.827**	0.705	0.591	0.415	0.516
satellite	0.949	0.884	**0.973**	0.970	**0.973**	0.903	0.904	0.938	0.746	0.857
kdd99	0.965	**1.000**	0.967	0.978	0.603	0.974	0.998	**1.000**	0.985	0.795
shuttle	**0.998**	**0.999**	0.974	0.975	0.467	0.992	0.996	0.995	0.882	0.017
speech	**0.500**	0.428	0.486	0.479	0.482	0.471	0.491	0.436	**0.625**	0.364
b-doorbell	0.971	**0.998**	0.997	**0.998**	0.754	0.972	0.935	0.997	**0.999**	0.889
b-thermostat	0.965	0.996	0.998	0.998	**0.999**	0.432	0.801	**0.999**	0.998	0.349
b-baby-monitor	0.957	0.989	0.973	0.985	0.444	0.861	0.905	**0.997**	**0.999**	0.871
b-security-camera	0.812	0.998	0.963	0.975	0.485	0.631	0.758	**1.000**	**1.000**	0.886
b-webcam	0.965	0.988	0.982	0.989	0.458	0.912	0.953	**0.997**	**0.998**	0.755
m-doorbell	0.995	0.999	**1.000**	**1.000**	**1.000**	0.993	0.989	**1.000**	**1.000**	0.894
m-thermostat	0.992	0.999	**1.000**	**1.000**	**1.000**	0.739	0.945	**1.000**	**1.000**	0.881
m-baby-monitor	0.974	**1.000**	**1.000**	**1.000**	**1.000**	0.951	0.964	**1.000**	**1.000**	0.883
m-security-camera	0.920	1.000	1.000	1.000	0.999	0.776	0.882	1.000	1.000	0.889

Table 3.4: Results from the comparative analysis: Average Precision. Highlighted in bold are the top two best values for each dataset. The upper section contains datasets from the *Goldstein* set, the lower one from the *N-BaIoT* set.

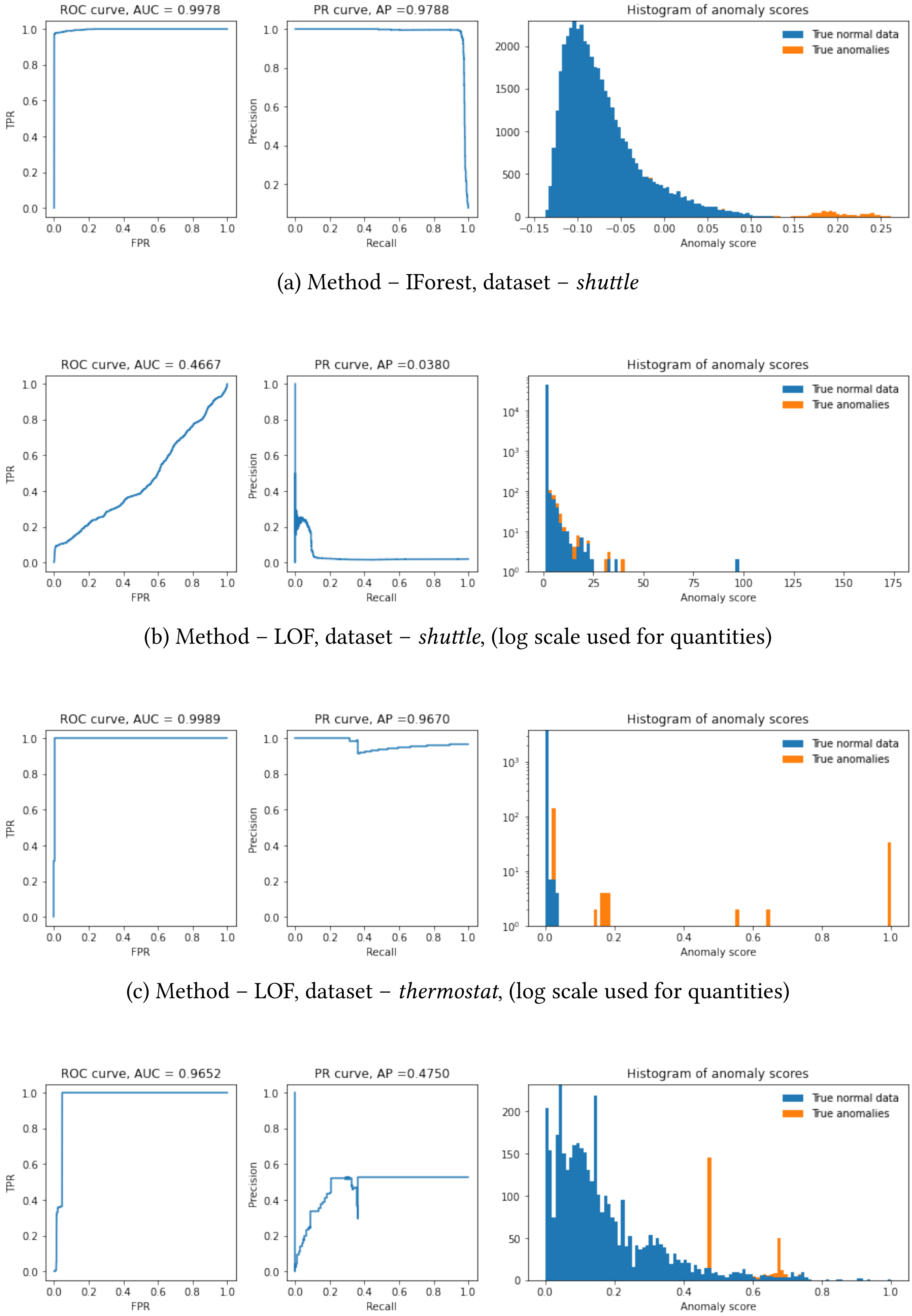
Dataset	IForest	OCSVM	kNN	kthNN	LOF	HBOS	COPOD	AE	Hybrid	SOGAAL
breast-cancer	0.733	**0.861**	0.595	0.618	0.675	0.520	0.715	**0.817**	0.108	0.017
pen-global	0.591	**0.970**	**0.833**	0.457	0.598	0.212	0.258	0.435	0.161	0.120
letter	0.085	0.160	**0.229**	0.175	**0.293**	0.077	0.068	0.120	0.066	0.047
aloi	0.033	**0.058**	0.049	0.043	**0.075**	0.028	0.031	0.036	0.043	0.034
pen-local	0.003	0.004	**0.046**	0.018	**0.110**	0.003	0.002	0.003	0.002	0.001
annthyroid	0.073	0.127	0.094	0.083	**0.165**	**0.140**	0.071	0.096	0.029	0.042
satellite	0.630	**0.688**	0.628	0.614	0.568	0.542	0.524	**0.675**	0.271	0.223
kdd99	0.526	**0.915**	0.203	0.226	0.004	0.335	**0.706**	0.666	0.184	0.004
shuttle	**0.979**	0.881	0.314	0.333	0.038	**0.956**	0.888	0.805	0.173	0.010
speech	0.017	0.016	0.019	0.019	0.020	**0.026**	0.019	0.016	**0.032**	0.012
b-doorbell	0.524	0.921	0.932	**0.936**	0.428	0.562	0.363	0.922	**0.967**	0.191
b-thermostat	0.475	0.860	0.947	0.950	**0.967**	0.057	0.118	**0.966**	0.958	0.084
b-baby-monitor	0.408	0.782	0.657	0.748	0.399	0.194	0.201	**0.926**	**0.977**	0.170
b-security-camera	0.152	0.939	0.615	0.672	0.412	0.083	0.089	**0.987**	**0.996**	0.189
b-webcam	0.474	0.744	0.712	0.779	0.336	0.274	0.328	**0.915**	**0.947**	0.097
m-doorbell	0.852	0.975	**1.000**	**1.000**	**1.000**	0.786	0.809	**1.000**	**1.000**	0.198
m-thermostat	0.759	0.958	**1.000**	0.995	**1.000**	0.092	0.363	**1.000**	**1.000**	0.181
m-baby-monitor	0.477	0.985	**1.000**	**1.000**	**1.000**	0.340	0.386	**1.000**	**1.000**	0.184
m-security-camera	0.361	0.998	0.999	**1.000**	0.999	0.098	0.395	**1.000**	**1.000**	0.192

lies from non-anomalous instances for datasets such as *m-doorbell* or *m-thermostat.* While the autoencoder or the hybrid method consistently produces excellent results for IoT attack data, the performance of other methods varies greatly across datasets. For example, the IForest algorithm, which works well with *shuttle* data, does not work so well with *b-thermostat* data. At the same time, though, LOF performs impressively on the *b-thermostat* dataset but fails to distinguish outliers in the *shuttle* data.

Figure 3.4 shows an additional illustration of selected algorithms performing differently on the same data. Whilst the IForest (F. T. Liu et al. 2008) method works very well for the *shuttle* dataset (very good separation of normal and anomalous data), the LOF (Breunig et al. 2000) method performs poorly over the same dataset. In contrast, on the *thermostat* dataset, LOF achieves clear separation of anomalies, whereas IForest blends the anomalous data with the normal instances. Although the high AUC value in the last case may suggest that the algorithm performed well, the high score is actually due to the large number of true negatives. The relatively poor precision reveals that the algorithm assigns the highest anomaly scores to normal data points, which is confirmed by the histogram of anomaly scores.

This observation, made on multiple datasets and confirmed by numerous evaluations throughout this study, is crucial to the research on selecting the best performing algorithm for a specific AD task. The methodology to address this problem, as outlined in *RQ1*, required data on the performance of various AD algorithms across diverse datasets. The remainder of this section describes how the performance data across all chosen datasets and algorithms was obtained.

Using the expanded framework for the algorithm selection problem presented in Section 1.2 and illustrated in Figure 1.1, the dataset space P was built on $N = 10,000$ datasets randomly chosen from the *Kandanaarachchi* set (Kandanaarachchi et al. 2020). Each dataset is defined as x_i where $i = 1, \ldots, N$. To generate the performance space Y, a selection of AD algorithms $\alpha_j \in A$ were evaluated over all the datasets in P. Two performance metrics, AUC and AP, were captured during the algorithm evaluation.



(a) Method – IForest, dataset – *shuttle*

(b) Method – LOF, dataset – *shuttle*, (log scale used for quantities)

(c) Method – LOF, dataset – *thermostat*, (log scale used for quantities)

(d) Method – IForest, dataset – *thermostat*

Figure 3.4: Examples of good and poor methods' performance of two selected methods: IForest and LOF with two datasets: *shuttle* and *thermostat*.

This experiment resulted in two matrices of the performance values $\mathbf{Y}_\nu \in \mathbb{R}^{N\times L}$, with $\nu \in \{\text{AUC}, \text{AP}\}$, where N and L denote the number of the datasets and the AD algorithms, respectively.

Procedure 1 AD algorithms performance data generation

```
Input:
    $A = \{\alpha_j \mid j = 1, \dots, L\}$
    $P = \{x_i \mid i = 1, \dots, N\}$
Output:
    $\mathbf{Y}_{\text{AUC}} \in \mathbb{R}^{N\times L}$
    $\mathbf{Y}_{\text{AP}} \in \mathbb{R}^{N\times L}$
for $j = 1$ to $L$ do
    for $i = 1$ to $N$ do
        $\left.\begin{matrix} y_{\text{AUC}\,ij} \\ y_{\text{AP}\,ij} \end{matrix}\right\} \leftarrow \alpha_j(x_i)$ { Run algorithm $\alpha_j$ over dataset $x_i$ }
    end for
end for
return $\mathbf{Y}_{\text{AUC}}, \mathbf{Y}_{\text{AP}}$
```

Procedure 1 outlines the steps taken to generate the evaluation values $\mathbf{Y}_{\text{AUC}}$ and $\mathbf{Y}_{\text{AP}}$ for the AD algorithm set A and the set of datasets P. The performance metrics obtained at this stage were made available via the IEEE public data repository (IEEEDataPort)[1] at http://ieee-dataport.org/10491. The AD algorithm selection, as one of the meta-learner components, is discussed in detail in Section 4.1.2.

3.4 Summary

The foundational elements of this study's experimental methodology – datasets, evaluation metrics, and generation of the algorithm performance data – lay the groundwork for the core experiments of this thesis, namely, the development and analysis of a meta-learner, as addressed in the research questions *RQ1*, *RQ2*, and *RQ3*. The next chapter builds upon this framework to propose and evaluate a meta-learner that can identify a suitable AD algorithm given a specific dataset.

[1] M. Gutowska, January 17, 2023, "Anomaly Detection Algorithms Performance", IEEE Dataport

Chapter 4

Meta-Learner

The lack of a single algorithm that works well across variable AD problems has been acknowledged by the academic community and is further confirmed by the results of the experiments presented in the previous chapter involving numerous AD algorithms and datasets. This gap motivates the development of a meta-learner capable of suggesting the appropriate algorithm. The motivation has been reinforced by the fact that the algorithm selection problem for unsupervised AD has, up to now, received very little attention from the research community. This problem has been captured within the research question: "Can an efficient meta-learner for unsupervised anomaly detection recommend the best algorithm for an unseen and unlabelled dataset?" (*RQ1*).

Two sections of this chapter, Section 4.1 and 4.2, refer to this problem. Section 4.1 presents the framework of the meta-learner developed in this study, and Section 4.2 outlines the evaluation methodology and discusses the obtained results.

The final section of this chapter (Section 4.3) analyses the influence of individual meta-learner components (i.e., meta-model, meta-features, set of base algorithms) on its final performance. This analysis responds to the research question: "Which components and design decisions of the meta-learner influence its overall performance?" (*RQ2*).

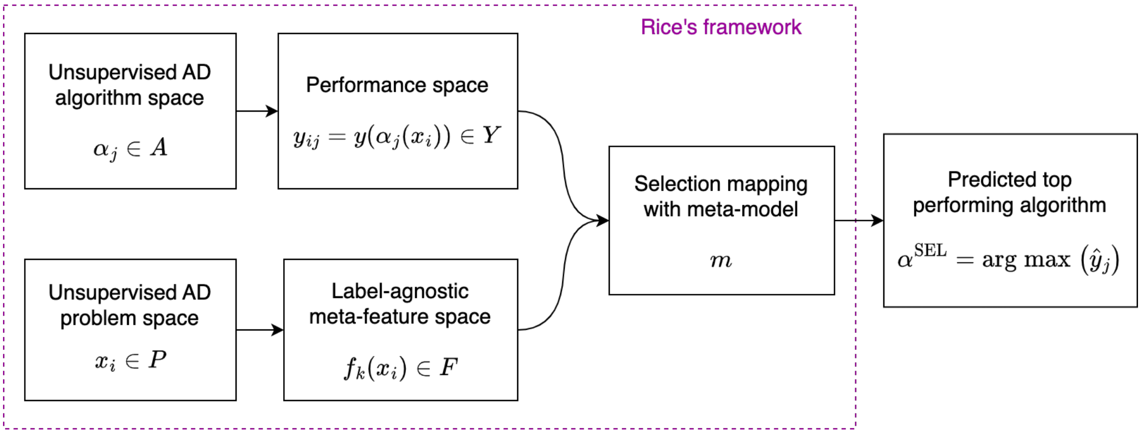

Rice's framework
Unsupervised AD algorithm space
$\alpha_j \in A$
Performance space
$y_{ij} = y(\alpha_j(x_i)) \in Y$
Unsupervised AD problem space
$x_i \in P$
Label-agnostic meta-feature space
$f_k(x_i) \in F$
Selection mapping with meta-model
m
Predicted top performing algorithm
$\alpha^{\mathrm{SEL}} = \arg\max\left(\hat{y}_j\right)$

Figure 4.1: Proposed meta-learner framework.

4.1 Meta-Learner Framework

The meta-learner proposed in this study adheres to Rice's concept (Rice 1976) of approaching the ASP. Figure 4.1 depicts its representation. In addition to the problem space P, described in Section 3.1, the meta-learner consists of three component parts: meta-features (F), a set of base AD algorithms (A), and the base learner – the meta-model (m). The following subsections elaborate on each component.

4.1.1 Meta-Features

The set F is comprised of 19 meta-features that were specifically designed to accommodate a broad range of anomaly characteristics. The overall design criteria for the features were as follows:

- to be independent of the data labels and therefore suitable for unsupervised scenarios,
- to capture the main types of anomalies, such as global, local, and collective,
- to describe multivariate characteristics of the data, i.e., mutual relations between the data points and their neighbourhood.

There were two steps involved in the process of feature generation:

1. calculation of distances between data points;
2. calculating the feature descriptors as statistical measures of the distance distributions.

In step 1, an approach inspired by Moran scatterplots depicting the relationship between global and local z-scores (Schubert et al. 2014) has been proposed to express characteristics related to local and global anomalies. The approach has been generalised to multivariate datasets by substituting the Mahalanobis distance for z-scores. Global and local Mahalanobis distances, H_G^l and H_{Ls}^l, were calculated for each data point, l, as expressed in Equations 4.1 and 4.2:

$$H_G^l(\overrightarrow{z_l}) = \sqrt{(\overrightarrow{z_l} - \overrightarrow{\mu})^T \mathbf{S}^{-1} (\overrightarrow{z_l} - \overrightarrow{\mu})} \tag{4.1}$$

$$H_{Ls}^l(\overrightarrow{z_l}) = \sqrt{(\overrightarrow{z_l} - \overrightarrow{\mu_s})^T \mathbf{S_s}^{-1} (\overrightarrow{z_l} - \overrightarrow{\mu_s})} \tag{4.2}$$

where $\overrightarrow{z_l} = (z_{1l}, ..., z_{Kl})$ represents a single observation, l, (a data point) with K features, $\overrightarrow{\mu} = (\mu_1, \mu_2, ..., \mu_K)$ represents the mean of all other observations in the dataset, and $\mathbf{S}$ is the covariance matrix. In Equation 4.2, $\overrightarrow{\mu_s}$ and $\mathbf{S_s}$ represent the mean and covariance matrix of the s nearest neighbours of $\overrightarrow{z_l}$, and H_{Ls}^l defines the Mahalanobis distance to the respective s nearest neighbours. The number of nearest neighbours has been chosen using a grid search approach and defined as $s \in \{20, 60, 80\}$. After this step, each data point, l, has been represented by four distances to the rest of the data: $\{H_G^l, H_{L20}^l, H_{L60}^l, H_{L80}^l\}$.

In step 2, the statistical characteristics of each of the distance profiles have been obtained. For each set of H_{Ls} and H_G the features such as TotalRange (TR), CenterMass (CM), TailHalf (TH), and TailQuarter (TQ) have been calculated as expressed in Equations 4.3–4.6:

$$\text{TR}_{H*} = \max(H_*^l) - \min(H_*^l) \tag{4.3}$$

$$\text{CM}_{H*} = \frac{1}{\text{TR}} \left(P^{75}(H_*^l) - P^{25}(H_*^l) \right) \tag{4.4}$$

$$\text{TH}_{H*} = \frac{1}{\text{TR}} \left(\max(H_*^l) - P^{50}(H_*^l) \right) \tag{4.5}$$

$$\text{TQ}_{H*} = \frac{1}{\text{TR}} \left(\max(H_*^l) - P^{75}(H_*^l) \right) \tag{4.6}$$

where H_* is one of H_G, H_{L20}, H_{L60}, H_{L80} and P^{25}, P^{75}, and P^{50} are the 25th and 75th percentiles and the median, respectively.

Procedure 2 Generation of distances for a dataset x_i (step 1)

```
Input:
    Dataset $x_i$ with observations $\{\mathbf{z}_l = (z_{1l}, \dots, z_{Kl})\}$
Output:
    Distances for each $\mathbf{z}_l$: $\mathbf{H} = \left\{H_G^l, H_{L20}^l, H_{L60}^l, H_{L80}^l \;:\; l = 1, \dots, n\right\}$
for $l = 1$ to $n$ do
    $H_G^l \leftarrow \sqrt{(\overrightarrow{z_l} - \overrightarrow{\mu})^T \mathbf{S}^{-1} (\overrightarrow{z_l} - \overrightarrow{\mu})}$ { Calculate global Mahalanobis distances }
    for all $s \in \{20, 60, 80\}$ do
        $H_{Ls}^l \leftarrow \sqrt{(\overrightarrow{z_l} - \overrightarrow{\mu_s})^T \mathbf{S_s^{-1}} (\overrightarrow{z_l} - \overrightarrow{\mu_s})}$ { Calculate local Mahalanobis distances }
    end for
end for
return H
```

Procedure 3 Generation of meta-features for a dataset x_i (step 2)

```
Input:
    $\mathbf{H} = \left\{H_G^l, H_{L20}^l, H_{L60}^l, H_{L80}^l \;:\; l = 1, \dots, n\right\}$
Output:
    Meta-feature vector $\mathbf{f} = (f_1, \dots, f_{19})$
$\mathbf{f} \leftarrow$ empty() { Initiate an empty feature vector }
for all $H_*^l \in \left\{H_G^l, H_{L20}^l, H_{L60}^l, H_{L80}^l\right\}$ do
    { Calculate statistics for distance distributions }
    $\mathrm{TR}_{H*} \leftarrow \max(H_*^l) - \min(H_*^l)$
    $\mathrm{CM}_{H*} \leftarrow \frac{1}{\mathrm{TR}} \left(P^{75}(H_*^l) - P^{25}(H_*^l)\right)$
    $\mathrm{TH}_{H*} \leftarrow \frac{1}{\mathrm{TR}} \left(\max(H_*^l) - P^{50}(H_*^l)\right)$
    $\mathrm{TQ}_{H*} \leftarrow \frac{1}{\mathrm{TR}} \left(\max(H_*^l) - P^{75}(H_*^l)\right)$
    Append $\left(H_G^l, H_{L20}^l, H_{L60}^l, H_{L80}^l\right)$ to $\mathbf{f}$
end for
for all $s \in \{20, 60, 80\}$ do
    $\mathcal{L}_s \leftarrow \frac{1}{n} \sum_l^n H_{Ls}^l / H_G^l$ { Calculate anomalies "locality" }
    Append $\mathcal{L}_s$ to $\mathbf{f}$
end for
return f
```

In addition, a property aiming to describe dataset "locality", $\mathcal{L}_s$, for each neighbourhood, s, was calculated using Equation 4.7:

$$\mathcal{L}_s = \frac{1}{n}\sum_{l}^{n} \frac{H_{Ls}^{l}}{H_{G}^{l}} \tag{4.7}$$

with n representing the total number of data instances within the dataset.

The steps of the meta-feature generation process for each dataset, x_i, are described in Procedure 2 (step #1) and Procedure 3 (step #2). These two procedures were performed for all the datasets from the problem space, P. The resulting features are summarised in Table 4.1.

Table 4.1: Meta-features proposed in this study.

Meta-feature		Instances	Feature Count
TR_{H*}	(TotalRange)	H_*: $\{H_G$, H_{L20}, H_{L60}, $H_{L80}\}$	4
CM_{H*}	(CenterMass)		4
TH_{H*}	(TailHalf)		4
TQ_{H*}	(TailQuarter)		4
$\mathcal{L}_s$	(Locality)	s: {20, 60, 80}	3
			Total: 19

The meta-feature set described aims to capture the characteristics of the primary anomaly types, namely global, local, and collective. This is achieved by including both global and local meta-features, the "locality" meta-feature, and by considering the nearest neighborhoods of each data point. Capturing of these characteristics was one of the initial design criteria. Additionally, another key design requirement involved capturing multivariate characteristics. This was addressed by treating the data points as objects in a multidimensional space, which contrasts with other common techniques that treat each data feature separately. Lastly, the creation of these meta-features did not require any labels, making this set fully suitable for unsupervised tasks.

4.1.2 Set of Anomaly Detection Algorithms

The base set of meta-learner algorithms, A, was selected to ensure representation from diverse families of methods, by including conventional or "classic" algorithms and modern deep learning-based techniques. In addition, the selected set contains the most widely used AD algorithms as implemented in practical applications or chosen by researchers (Campos et al. 2016; Emmott et al. 2013; Goldstein and Uchida 2016; Kandanaarachchi et al. 2020; H. Wang et al. 2019). It also incorporates the algorithms employed by the state-of-the-art, MetaOD (Zhao, Rossi, et al. 2021). The set includes ten conventional and three neural network-based AD algorithms as listed in Table 4.2.

Table 4.2: AD models and their parameters comprising the set used in this study.

AD algorithm	Parameter 1	Parameter 2
LOF	n_neighbors = 60	distance = 'euclidean'
kNN	n_neighbors = 60	method = 'mean'
k^{th}NN	n_neighbors = 60	method = 'largest'
OCSVM	nu = 0.008	kernel = 'rbf'
COF	n_neighbors = 60	N/A
ABOD	n_neighbors = 60	N/A
IForest	n_estimators = 100	max_features = 1.0
HBOS	n_bins = 90	tolerance = 0.5
COPOD	N/A	N/A
PCA	n_components = 'mle'	svd_solver = 'full'
VAE	epochs = 500	hidden layers, as described in Eq. 4.8
SO-GAAL	epochs = 25	N/A
MO-GAAL	epochs = 25	N/A

To cover a diverse range of algorithm families, the "classic bucket" involved algorithms from the following categories:

- Density-based methods – Local Outlier Factor (LOF) (Breunig et al. 2000), Connectivity-Based Outlier Factor (COF) (Tang et al. 2002),
- Distance-based methods – k-Nearest Neighbours (kNN) (Ramaswamy et al. 2000), Angle-based Outlier Detector (ABOD) (Kriegel et al. 2008),
- Classification-based methods – One-Class Support Vector Machines (OCSVM) (Schölkopf et al. 2000),
- Projection-based methods – Isolation Forest (IForest) (F. T. Liu et al. 2008),
- Statistical-based methods – Histogram-based Outlier Score (HBOS) (Goldstein and Dengel 2012), Copula-based Outlier Detection (COPOD) (Z. Li et al. 2020),

- Subspace-based methods – Principal Component Analysis-based anomaly detection (PCA) (Shyu et al. 2003).

Deep learning-based approaches included Variational Autoencoder (VAE) (Kingma and Welling 2013), Single-Objective Generative Adversarial Active Learning (SO-GAAL), and Multi-Objective Generative Adversarial Active Learning (MO-GAAL) (Y. Liu et al. 2019).

Table 4.2 presents the chosen algorithms and their parameters. The optimal parameters were chosen as the best ones by averaging across numerous datasets. The VAE's architecture was designed individually for each dataset. The number and dimensions of the hidden layers were determined by the number of features in each dataset. Given that K represents the number of features in a given dataset, the number of nodes in each hidden layer was set as described in Equation 4.8.

$$\begin{cases} K \times [0.75, 0.5, 0.33, 0.25] & \texttt{if} \quad K \geq 100 \\ K \times [0.75, 0.5, 0.25] & \texttt{if} \quad 100 > K \geq 50 \\ K \times [0.5, 0.25] & \texttt{if} \quad 50 > K \geq 6 \\ 2 & \texttt{if} \quad K < 6 \end{cases} \tag{4.8}$$

The implementation of these algorithms from the Python PyOD package (Zhao, Nasrullah, et al. 2019) was used to perform their evaluation.

4.1.3 Meta-Model

A key part of a meta-learner is a meta-model, whose goal is to select the AD algorithm that performs best on an unseen and unlabelled AD dataset. The meta-model, m, proposed in this study is based on a neural network architecture. Neural networks have proven successful in a variety of tasks and are relatively time-efficient when training for small-size problems. The capability to efficiently perform a multi-factor regression is an additional benefit, particularly when a simultaneous response for an array of algorithms is expected.

The architecture proposed in this work features three hidden layers (64, 64, and 32 nodes), a dropout of 0.2 after each layer, and the predictive regression layer that outputs the predicted performance values $\hat{y}$ for each algorithm $\alpha_1, \dots, \alpha_L$, with L being the number of algorithms used in the meta-training. Figure 4.2 depicts the meta-model framework, including the multi-factor response.

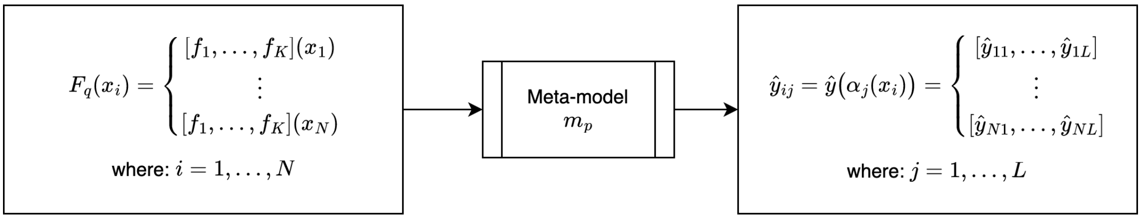

$F_q(x_i) = \begin{cases} [f_1, \dots, f_K](x_1) \\ \vdots \\ [f_1, \dots, f_K](x_N) \end{cases}$
where: $i = 1, \dots, N$
Meta-model
m_p
$\hat{y}_{ij} = \hat{y}(\alpha_j(x_i)) = \begin{cases} [\hat{y}_{11}, \dots, \hat{y}_{1L}] \\ \vdots \\ [\hat{y}_{N1}, \dots, \hat{y}_{NL}] \end{cases}$
where: $j = 1, \dots, L$

Figure 4.2: Regression meta-model framework with its multi-factor response.

The meta-model training was conducted using a typical supervised machine-learning pipeline, where the datasets, x_i, were split in a 60:15:25 ratio into the train, validation, and test sets. The meta-model was trained to find a mapping between the meta-features and the AD algorithm performance metrics. The validation set was used to inform the decision on when to end the training process. The learnt mapping was then applied to the set of test datasets, x_i^{test}, where the performance values of each AD algorithm, $\hat{y}_{ij}$, were predicted.

Batches of 32 samples were used in training, and the Adam optimiser (Kingma and Ba 2014) was employed to minimise the mean squared error. The network architecture parameters, such as the number of hidden layers, the number of nodes, dropout level, epoch count, and batch size, were optimised with the *Weights and Biases* tool (Biewald 2020).

4.2 Meta-Learner Evaluation

The evaluation of the meta-learner is performed as follows. The meta-model predicts performance values, $\hat{y}_{ij}$, for each dataset, x_i, and each algorithm, α_j. Following this, an algorithm, α^{SEL}, is selected from the set of algorithms, $\alpha_j \in A$, for each dataset, x_i, to maximise the predicted performance, $\hat{y}_{ij}$. The actual performance, y_i^{SEL}, of the algorithm, α^{SEL}, obtained on dataset, x_i, serves as the performance metric for evaluating the efficacy of the meta-learner on the given dataset. The formulation of this approach can be expressed using the equations:

$$\hat{y}_{ij} := \hat{y}_i(\alpha_j) = m\big(F(x_i)\big) \tag{4.9}$$

$$y_i^{\mathrm{SEL}} := y_i(\alpha^{\mathrm{SEL}}) : \alpha^{\mathrm{SEL}} = \arg\max(\hat{y}_{ij}) \tag{4.10}$$

Since the meta-learner was assessed on the AUC and AP metrics, the y variable in the above equations can be either a value of AUC or an AP. The choice of y_i^{SEL} as a meta-learner performance measure can be justified by the fact that such a metric is proportionally greater as the selection gets closer to the real best performing method.

Another metric used in this study to evaluate a meta-learner performance was its error, D_i, which, for the dataset x_i, has been defined as the *Distance from the Top*, the difference between the best measured performance, y_i^{TOP}, and the measured performance of the algorithm selected by meta-learner, y_i^{SEL}, as shown in Equation 4.11:

$$D_i = y_i^{\mathrm{TOP}} - y_i^{\mathrm{SEL}}. \tag{4.11}$$

This metric assists in determining how much the method suggested by the meta-learner differs from the highest performing algorithm for a particular dataset. It is noteworthy to mention that this error provides insight into the "virtual best" performance on a given dataset through quantifying the discrepancy between the suggested and the optimal algorithm performance.

To support the selection of these metrics, it should be noted that both y_i^{SEL} and D_i are rank-independent and easily interpretable because they are on the same scale as actual AUC or AP values.

4.2.1 Evaluation Methodology

The performance of the proposed meta-learner has been compared with the present state-of-the-art solution, MetaOD, as proposed by Zhao, Rossi, et al. 2021. The authors of MetaOD did not explicitly reference Rice's framework, but their method is similar in that it includes a set of meta-features, a set of base AD algorithms, and a meta-model. As previously stated in Chapter 2, their algorithm selection strategy was based on collaborative filtering (CF) and made use of the matrix factorisation method.

The subsequent sections of this chapter make extensive use of both methodologies, so the indices *1* and *2* are introduced when referring to the meta-learner components (meta-model, meta-features, algorithms) proposed in this work (m_1, F_1, A_1) and MetaOD (m_2, F_2, A_2), respectively. Table 4.3 briefly compares the settings of the two approaches. The indexes, p, q, and r have been introduced for effective iteration on these components and are leveraged mainly in Section 4.3.

Table 4.3: Summary of the components of meta-learners compared in this study.

	Current work ($p, q, r = 1$)	MetaOD ($p, q, r = 2$)
Meta-model m_p	Neural Network (NN)	Collaborative Filtering (CF)
Meta-features F_q	19 features specific to AD problems	200 features, combined: statistical and landmarking features
Set of AD algorithms A_r	13 distinct algorithms	298 models: 8 algorithms combined with sets of hyperparameters

The mean meta-learner performance $\overline{y}^{\mathrm{SEL}}$ measured as AUC and AP over the set of test datasets, and the meta-learners' mean error $\overline{D}$ have been compared. The statistical significance has been measured using the *Paired t-Test* (paired difference test). In addition to statistical significance, the practical significance (effect size) has been assessed using Cohen's d (Cohen 2013) as outlined in Equations 4.12 and 4.13:

$$d_y = \frac{\overline{y}_1^{\mathrm{SEL}} - \overline{y}_2^{\mathrm{SEL}}}{s_y^*} \tag{4.12}$$

$$d_D = \frac{\overline{D}_1 - \overline{D}_2}{s_D^*}, \tag{4.13}$$

where subscripts *1* and *2* indicate approaches from this study and MetaOD, respectively, and s_y^* and s_D^* are the pooled standard deviation of *1* and *2* distributions of performance values and errors, respectively. The use of the effect size was motivated by the large sample sizes. The number of observations in such cases makes the variables appear

Table 4.4: Comparison of the mean performance and the mean error with standard deviations across test datasets of two analysed meta-learners.

		Current study	MetaOD	D.f.	T-stat.	p-value	Effect size
AUC	$\overline{y}^{\mathrm{SEL}}$	**0.6703** $\pm$ 0.1820	0.6464 $\pm$ 0.1940	2317	7.882	<0.001	0.126
	$\overline{D}$	**0.1567** $\pm$ 0.1315	0.1804 $\pm$ 0.1558	2317	-7.882	<0.001	0.162
AP	$\overline{y}^{\mathrm{SEL}}$	**0.1413** $\pm$ 0.1955	0.1369 $\pm$ 0.1905	2302	2.193	0.028	0.009
	$\overline{D}$	**0.1685** $\pm$ 0.1466	0.1749 $\pm$ 0.1543	2302	-2.193	0.028	0.042

D.f. – degrees of freedom, T-stat. – t-test statistic

statistically significant. As a result, practical significance is a more useful statistic to recognise. The strength of an effect can be categorized as follows (Cohen 2013):

$$\text{small effect} \leq 0.2 < \text{medium effect} \leq 0.5 < \text{large effect}.$$

4.2.2 Results and Discussion

The performance comparison of the two solutions for the AUC and AP-based experiments is presented in Table 4.4. The mean values of performance $\overline{y}^{\mathrm{SEL}}$ and the error $\overline{D}$ are obtained over the test set of datasets x_i^{test}. Better performing solutions are highlighted with bold font. In both cases, AUC and AP, the difference in the mean performance and the mean error between the solution proposed in the current research and MetaOD is statistically significant ($p < 0.001$ for AUC, $p = 0.028$ for AP).

The results, however, demonstrate that while there was a statistically significant improvement with the proposed method, there was a negligible effect (practical difference) when comparing the AP mean error (Cohen's d = 0.042) and a small effect when comparing the AUC mean error (Cohen's d = 0.162) between the two approaches. Therefore, while an improvement was observed, it is important to note that its magnitude is not substantial. The proposed method in this study demonstrates, however, that equivalent results can be obtained from a substantially reduced feature set and the omission of hyperparameter optimisation (HO) in the meta-learning configurations. Previous works have assumed that HO was the main characteristic in meta-learning (Feurer, Springenberg, et al. 2015; Horváth et al. 2016; Komer et al. 2014).

While the above early findings are only a direct comparison of the approach described in the current study with the MetaOD, the analysis offered in Section 4.3 helps to understand the meta-learner characteristics that contribute to its success.

4.2.3 Time Efficiency Analysis

Aside from the accuracy performance assessment, the time efficiency of both strategies was compared. The time analysis was performed on a subset of 20 datasets chosen at random from the entire set used in the current study. The datasets ranged in size from 68 to 5,186 observations and 5 to 147 features. This analysis has been restricted to the end-user perspective, which includes the generation of dataset meta-features and the prediction of the best-suited algorithm. The time summary of both approaches to generate meta-features and perform prediction is presented in Table 4.5 and Table 4.6, respectively.

Table 4.5: Time in seconds to generate dataset meta-features summarised for a random sample set of 20 datasets.

Statistic (time, s)	MetaOD	Current study
Mean	0.886	2.265
St. dev.	0.777	3.627
Min	0.351	0.044
Max	3.103	10.517
Cases with shorter time	8	12

Table 4.6: Time in seconds to predict the best performing algorithm summarised for a random sample set of 20 datasets.

Statistic (time, s)	CF	NN
Mean	1.376	0.492
St. dev.	0.574	0.114
Min	0.914	0.400
Max	4.537	0.956

Although it takes less time on average for the MetaOD to generate the meta-feature set, the number of datasets for which the generation takes less time is greater for the currently presented approach. The current approach is more time-consuming for datasets with a relatively higher number of observations because it involves calculating the distances between all instances within a dataset. Ultimately, the time difference between MetaOD and the current approach was not statistically significant for the measured sample set ($t_{19} = 1.835$, $p = 0.082$). When compared to the CF meta-model, the prediction times are shorter on average and more consistent with the use of NN. Furthermore, training times for CF models were significantly longer than for NN-based models (approx. 20 hours versus approx. 10 minutes per meta-learner variant). The exact measures are not included because the meta-learners' training was conducted in parallel on machines with varying capabilities, so a precise comparison of the training times was not possible.

The presented results were obtained on the machine with the following subcomponents: 1.6 GHz Dual-Core Intel Core i5 processor and 8 GB of 2133 MHz RAM.

Overall, the results of the time analysis demonstrate that using a meta-learner within an AD pipeline outweighs the costs in terms of time and computing resources. For a dataset with 1,000 observations and 45 features, the extra time of 1-2 seconds for meta-feature generation and 0.5 seconds for finding the best suited algorithm could potentially save hours on a trial-and-error process of finding the best performing algorithm and evaluating the results.

4.3 Contribution of Meta-Learner Components

Section 4.1, which details the components involved in constructing the meta-learner for unsupervised AD, illustrates the importance of the decision-making process. The design choices made during the development of the meta-learner considerably influence its ultimate performance. In light of the absence of research examining the impact of such design decisions, the present study has been undertaken aiming to address this gap. This problem is captured in the research question *RQ2* ("Which components and design decisions of the meta-learner influence its overall performance?").

The current section presents the approach taken to address the *RQ2*. Section 4.3.1 introduces the experimental design employed, Section 4.3.2 presents the analytical method-

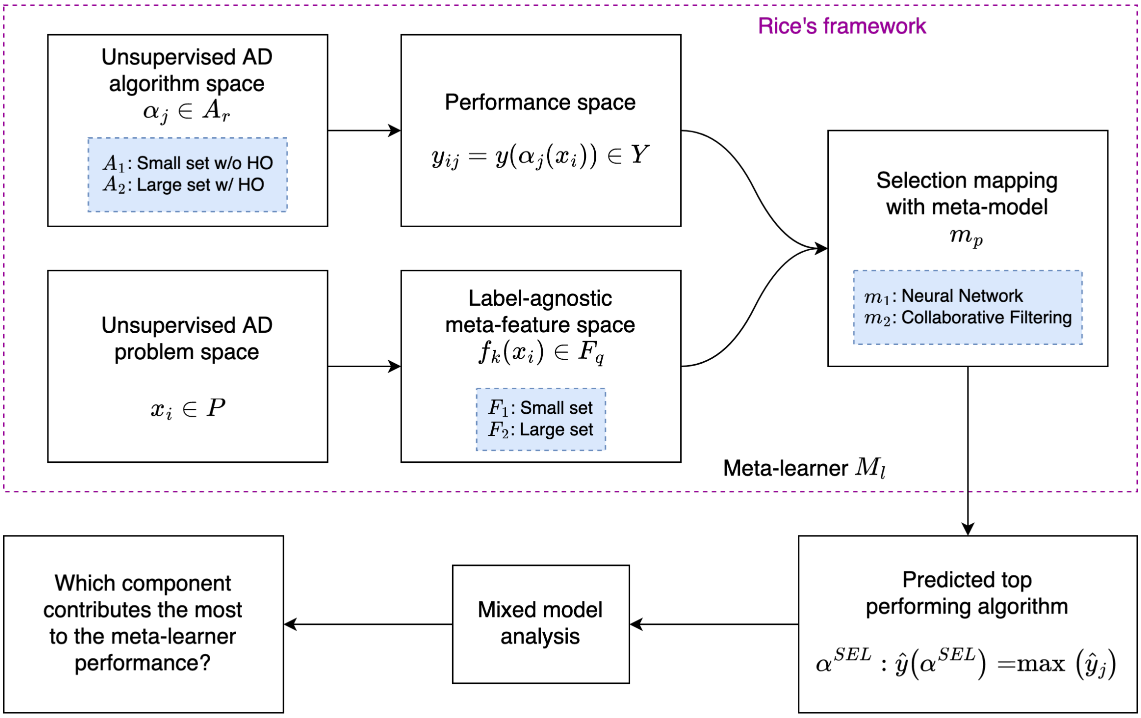

Rice's framework
Unsupervised AD algorithm space
$\alpha_j \in A_r$
A_1: Small set w/o HO
A_2: Large set w/ HO
Performance space
$y_{ij} = y(\alpha_j(x_i)) \in Y$
Selection mapping with meta-model
m_p
m_1: Neural Network
m_2: Collaborative Filtering
Unsupervised AD problem space
$x_i \in P$
Label-agnostic meta-feature space
$f_k(x_i) \in F_q$
F_1: Small set
F_2: Large set
Meta-learner M_l
Which component contributes the most to the meta-learner performance?
Mixed model analysis
Predicted top performing algorithm
$\alpha^{SEL} : \hat{y}(\alpha^{SEL}) = \max(\hat{y}_j)$

Figure 4.3: Experimental design including the meta-learner framework with reference to Rice's representation (outlined with a purple line) and factor components used in comparative evaluation (blue stickers).

ology, and Section 4.3.3 concludes this research problem by presenting and discussing the results.

4.3.1 Experimental Design

This stage of the experiment employs a 2^3-factorial design. The factor components selected for the analysis – including the meta-feature generation strategies F_q, the base set of AD algorithms A_r, and the meta-model m_p – are drawn from the methodology proposed in the present study and the MetaOD (Zhao, Rossi, et al. 2021). These components have been described in Table 4.3, in Section 4.2. The experimental design, an extended version of the meta-learner framework (from Figure 4.1), is illustrated in Figure 4.3.

To perform the experiment, eight meta-learners $M_l := M_{pqr}$, were designed, where $l = 1, \ldots, 8$, and $p, q, r \in \{1, 2\}$, which incorporated two variants of mentioned factors F_q, A_r, and m_p. Indices *1* and *2* have been used to denote factors from the current study and MetaOD, respectively. Each combination of the 2^3-factorial design was implemented on each candidate dataset x_i producing the predicted performance metrics $\hat{y}_{ij}$

for each algorithm α_j. The performance metrics of meta-learners' selected algorithms $\alpha^{\text{SEL}} \in A$ were then analysed using a mixed model analysis, where x_i was considered to be the subject. Similarly to the approach described in Section 4.2, the algorithm α^{SEL} was selected from $\alpha_j \in A$ for each x_i to maximise the predicted performance $\hat{y}_{ij}$. The steps performed are described in Procedure 4 and 5. The Equations 4.9 and 4.10 from Section 4.2 remain applicable, but they at this point apply to all $p, q, r \in \{1, 2\}$, where:

$$\begin{cases} m & = & m_p \\ F & = & F_q \\ \alpha_j & \in & A_r \end{cases}$$

The architecture of all NN-based variants followed the one described in Section 4.1.3. The training of each variant has been started from random weights and run through up to 1000 epochs. The *Early Stopping* functionality from Keras library (Chollet et al. 2015) has been implemented to cease further training when no improvement was observed, as measured by the loss on the validation data. Subsequently, the optimal weights have been obtained based on the validation loss. The resulting training length ranged for different meta-learner variants from 300 to 1000 epochs. The batches of 32 samples were used in training, and the Adam optimiser (Kingma and Ba 2014) was employed to minimise the mean squared error.

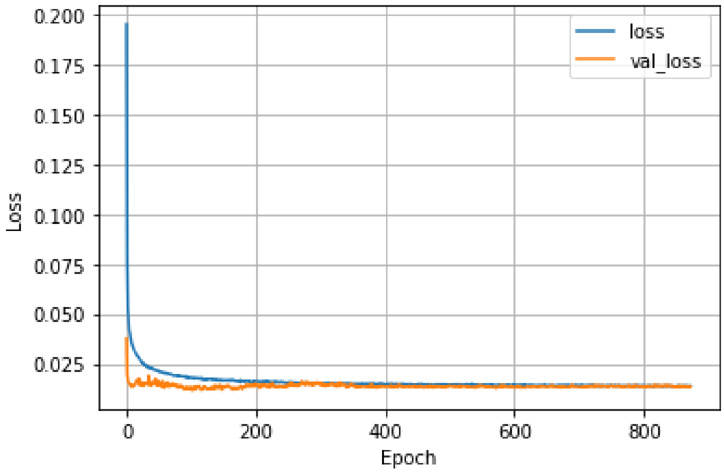


Figure 4.4: Training loss (*loss*) and validation loss (*val_loss*) versus epochs in training one of the NN-based meta-model variants.

Figure 4.4 presents an example of the training process picturing the training and validation loss in subsequent epochs. As expected, both training and validation loss are lowering. The fluctuations in the validation curve are due to a relatively small amount of validation data, but the overall trend is diminishing, implying a stable minimum on the loss surface.

All architecture and training hyperparameters, such as the number of hidden layers, the number of nodes, dropout level, epoch count, and batch size, were chosen using the grid search approach and optimised with the *Weights and Biases* tool (Biewald 2020). The search for the optimal architecture ranged from 2-layered networks of 4 + 4 nodes to 3-layered networks of 64 + 64 + 64 nodes. More complex architectures were not considered to avoid the risk of model overfitting.

Procedure 4 Training of meta-learners M_{prq}

```
Input:
    $\mathbf{F}_q \in \mathbb{R}^{N \times K_q}$
    $\mathbf{Y}_{r\,\text{AUC}}, \mathbf{Y}_{r\,\text{AP}} \in \mathbb{R}^{N \times L_r}$, where $\mathbf{Y}_r = Y(A_r)$
Output:
    $Y^{\text{SEL}}_{\text{AUC}}, Y^{\text{SEL}}_{\text{AP}} \in \mathbb{R}^{N^{\text{test}}}$
for all $\nu \in \{\text{AUC}, \text{AP}\}$ do
    for all $F_q : q \in \{1, 2\}$ do
        for all $Y_r : r \in \{1, 2\}$ do
            for all $m_p : p \in \{1, 2\}$ do
                { Split the input and output data }
                $(F_q^{\text{train}}, F_q^{\text{test}}) \leftarrow F_q$
                $(Y_r^{\text{train}}, Y_r^{\text{test}}) \leftarrow Y_r$
                $m_p \leftarrow \text{train}(F_q^{\text{train}}, Y_r^{\text{train}})$ { Train the meta-model in a supervised manner }
                $\widehat{\mathbf{Y}}_r^{\text{test}} \leftarrow m_p(F_q^{\text{test}})$ { Predict }
                Procedure 5 { Select the best algorithm and obtain its performance }
            end for
        end for
    end for
end for
return $Y^{\text{SEL}}_{\text{AUC}}, Y^{\text{SEL}}_{\text{AP}}$
```

Procedure 5 Find performance of the best predicted algorithm

```
Input:
    $\widehat{\mathbf{Y}}^{\mathtt{test}} \in \mathbb{R}^{N^{\text{test}} \times L}$
Output:
    $Y^{\text{SEL}} \in \mathbb{R}^{N^{\text{test}}}$
for $i = 1$ to $N^{\text{test}}$ do
    $y_i^{\text{SEL}} \leftarrow y_i(\alpha^{\text{SEL}})\ :\ \alpha^{\text{SEL}} = \arg\max(\hat{y}_{ij})$
end for
return $Y^{\text{SEL}}$
```

The training of the CF-based meta-models has been done according to the methodology described in the work by Zhao, Rossi, et al. 2021 and the instructions from the code library (Zhao, Rossi, et al. 2020).

4.3.2 Statistical Analysis

This section presents the mixed-model analysis (Demidenko 2013; McCulloch et al. 2004; Stroup 2012) that was carried out to examine the factors $Z_c \in \{m_p, F_q, A_r\}$ with $c = 1, \ldots, 3$, contributing to the error of the meta-learners in choosing the best-performing algorithm.

The error of the meta-learner used in this analysis is the one defined in Equation 4.11, however, it ranges now through all meta-learner variants $M_l : l = 1, \ldots, 8$ and datasets $x_i : i = 1, \ldots, N$ as outlined in Equation 4.14:

$$D_{il} = \ln\left(y_i^{\text{TOP}} - y_{il}^{\text{SEL}}\right). \tag{4.14}$$

The $\ln(\cdot)$ operation has been used to normalise the error distribution.

Thirty principal components $V_{1i}, \ldots, V_{ni}$ (which explain 93 % of the variance) were generated from concatenated sets of meta-features F_1 and F_2 and used as covariates to adjust for variability due to the differences between datasets.

The model used is expressed in Equations 4.15–4.17:

$$D_{il} = \beta_0 + \beta_1 Z_1 + \beta_2 Z_2 + \beta_3 Z_3 + \beta_{1'} V_{1i} + \cdots + \beta_{n'} V_{ni} + \gamma_{0i} + \varepsilon_{il} \tag{4.15}$$

with

$$\gamma_{0i} \sim N\left(0, \sigma_\gamma^2\right) \tag{4.16}$$

$$\varepsilon_{il} \sim N\left(0, \sigma_\varepsilon^2\right) \tag{4.17}$$

where $\beta_0, \ldots, \beta_{n'}$ describe the fixed effects (Stroup 2012), and γ_{0i} expresses the random effects' intercepts. Fixed coefficients $\beta_0, \ldots, \beta_3$ were assessed for significance using the F-test with statistical significance set at $p < 0.05$. The effect size (Cohen's d) of the three components Z_c was calculated using the following:

$$d_c = \frac{\beta_c}{\sqrt{\sigma_x^2 + \sigma_\varepsilon^2}}, \tag{4.18}$$

where $c = 1, \ldots, 3$ and β_c represents the fixed parameters estimates of Z_c, as in Equation 4.15, and σ_x^2 and σ_ε^2 represent the variance of the random components and the error term, respectively, as expressed in Equations 4.16 and 4.17.

4.3.3 Results and Discussion

Table 4.7 displays a performance summary of eight meta-learners using mean performance across test datasets, x_i^{test}. It's worth noting that the hybrid technique of F_2, A_2, and m_1 yields the greatest performance results for both meta-learner series, AUC, and AP-based. Another interesting finding is that meta-learners that employ the m_1 meta-model outperform all CF-based learners. The subsequent component analysis provides a more in-depth understanding of these observations.

The mixed model analysis, summarised in Table 4.8, shows that whereas the choice of each component, Z_c, is statistically significant for AUC ($p < 0.001$), the larger meta-feature set, F_2, provides only a marginal benefit over the smaller set of meta-features, F_1, (Cohen's d = 0.082). It is worth noting that while F_2 makes extensive use of generic sta-

Table 4.7: Mean performance $\overline{y}^{\text{SEL}}$ with standard deviations across test datasets of eight meta-learners for AUC and AP.

Meta-model	AD alg. set	Meta-features	$\overline{y}^{\text{SEL}}$, AUC	$\overline{y}^{\text{SEL}}$, AP
m_1 (NN)	A_2	F_2	**0.692** ± 0.177	**0.154** ± 0.215
m_1 (NN)	A_2	F_1	0.679 ± 0.179	0.150 ± 0.209
m_1 (NN)	A_1	F_2	0.678 ± 0.183	0.146 ± 0.200
m_1 (NN)	A_1	F_1	0.670 ± 0.182	0.141 ± 0.195
m_2 (CF)	A_2	F_2	0.646 ± 0.194	0.137 ± 0.191
m_2 (CF)	A_2	F_1	0.630 ± 0.202	0.136 ± 0.192
m_2 (CF)	A_1	F_2	0.630 ± 0.191	0.127 ± 0.176
m_2 (CF)	A_1	F_1	0.616 ± 0.193	0.120 ± 0.169

tistical features, the compact set, F_1, is crafted to reflect anomaly characteristics. When comparing differences in the performance between the two sets of AD algorithms, the large set with HO (A_2) outperforms the small set without HO (A_1). However, given the number of models in both groups (298 versus 13), the effect size is not as compelling as one would expect (Cohen's d = 0.130). The largest effect size associated with the choice of meta-model reveals that it has the greatest impact on the meta-learner's ultimate performance, with the NN-based meta-model, m_1, outperforming the state-of-the-art CF approach, m_2 (Cohen's d = 0.300).

This outcome is an important consideration given that current AutoML or meta-learning studies frequently direct their attention to other aspects, such as meta-features development (Kanda et al. 2016; Kotlar et al. 2021) or HO (Feurer, Springenberg, et al. 2015; Horváth et al. 2016; Komer et al. 2014). The findings of this analysis reveal that, given the cost of pre-evaluating a large number of algorithms, the comprehensive grid search strategy over feasible algorithms and hyperparameters is not very beneficial in the examined context. This work highlights the significance of underexplored aspects in meta-learning, particularly in the context of unsupervised AD, which play a crucial role in influencing the overall results. The research demonstrates that putting time and effort into developing an adequate meta-model that can effectively utilise data from previous evaluations is the most promising way of improving meta-learners for unsupervised AD.

The contributions measured on AP follow an analogous pattern, but the impacts are more subtle. Because of the AP metric's "lower resolution" (highly skewed distribu-

Table 4.8: Type III analysis of the main effects of the meta-learner components.

	Component	D.f.	F-stat.	p-value	Effect size
AUC	Intercept	1,2291	13382.78	<0.001	
	Meta-features F_q	1,16244	51.457	<0.001	0.082
	AD models A_r	1,16244	127.648	<0.001	0.130
	Meta-model m_p	1,16244	678.938	<0.001	**0.300**
AP	Intercept	1,2294	14091.410	<0.001	
	Meta-features F_q	1,16009	0.139	0.710	0.004
	AD models A_r	1,16025	69.087	<0.001	0.095
	Meta-model m_p	1,16016	111.051	<0.001	0.121

D.f. – degrees of freedom, F-stat. – F-test statistic

tions: skewness – 2.647, kurtosis – 7.064), the discrepancy is less apparent. The skewed distribution of AP values is to be expected because unbalanced datasets often exhibit this behaviour (Haibo et al. 2013; Viola et al. 2022). The influence of the meta-features is not statistically significant at 0.05 significance level. Consequently, the effect size is negligible. The other two components contribute more, but their effect sizes on the AP metric are also minimal. Nonetheless, the NN meta-learner demonstrated a favourable performance in comparison to the CF group (Table 4.7).

The interaction terms between the main effects outlined in Equation 4.15 were initially examined but had no statistical significance and were subsequently excluded from the final statistical model.

The observations and findings presented above addressed the research question *RQ2*, which concerned the impact of meta-learner components on its overall performance. It is worth noting that the statistical analysis and formulation of a fresh perspective were made possible by the use of a large range of AD benchmark datasets. Such an approach would not be possible with usual sets, which are commonly utilised in other studies addressing the ASP problem. As such, this section also extends to the research question *RQ4* addressing the use of the largest AD benchmark dataset currently available.

Chapter 5

Risk Assessment Strategy

The preceding chapter described the architecture of a meta-learner developed in the current study, inspecting the influence of its distinct components on overall performance. In contrast, this chapter shifts its focus towards assessing the quality and reliability of individual responses given by the meta-learner.

As highlighted in Sections 1.3 and 2.4, the evaluation of the reliability of individual predictions generated by machine learning (ML) models remains an underexplored area of study. While there are several studies on this topic discussing ML models in general, there is a notable lack of studies on localised evaluation in the context of meta-learning. The emphasis on individual predictions is considerably more important for meta-learning, where each meta-learner's output corresponds to a distinct problem. For practitioners, the reliability of a response relevant to their case is more insightful and actionable than a meta-learner's collective performance across multiple tasks and domains.

Recognising both the importance of this subject and the existing research gap, the research presented in this chapter aims to make a substantive contribution by examining the meta-feature space – the space formed on the meta-learner's input features. It specifically seeks to identify areas where the likelihood of erroneous meta-learner outputs is higher than in other regions and to quantify that likelihood. This problem is encapsulated in the research question: "Can the reliability of individual meta-learner responses be evaluated and high-risk areas within the meta-feature space be identified?" (*RQ3*), as outlined in Section 1.3.

The term *risk*, as employed in this research, describes the likelihood of receiving suboptimal responses from the meta-learner. Given that this likelihood is localised, pertaining specifically to individual responses of the meta-learner, the concept of risk also applies to particular regions within the meta-feature space that demonstrate less stability and are, consequently, considered high-risk. To address the presented problem comprehensively, this research adopts an approach that consists of two phases:

- quantification of the risk of the meta-learner's algorithm recommendations being less than optimal, using the proposed Risk metric,
- probabilistic estimation of the upper bound of the meta-learner errors for individual responses.

The remainder of this chapter is organised as follows: Section 5.1 formulates the problem and introduces the essential terminology, the approach adopted to tackle the given problem is described in Section 5.2, and the findings along with the accompanying discussion are presented in Section 5.3.

5.1 Problem Formulation

The two-fold objective of this research entails splitting the problem into two distinct parts:

1. Given a new, unlabelled dataset and the AD algorithm then recommended by the meta-learner, can we quantitatively determine the *risk* that the suggested algorithm will not be the optimal choice for this particular dataset?
2. Given the *risk*, corresponding to the region of the meta-feature space in which the dataset is located, can we determine the maximum amount by which the recommended algorithm will underperform compared to the true best algorithm for the dataset?

Let the $\textsc{Risk}_{x_i}$ be defined as a measure of the likelihood that the meta-learner response for the dataset x_i results in performance that is less than that of the optimal algorithm. It is postulated that Risk can be calculated as a function of the dataset's meta-features:

$$\textsc{Risk} = g(f_1, \ldots, f_K) \tag{5.1}$$

For each dataset, x_i, consider the meta-learner's associated error, D_i, defined in Chapter 4 with Equation 4.11, and referred to as the *Distance from the Top* – the difference between the best measured AD algorithm performance, y_i^{TOP}, and the performance of the algorithm selected by the meta-learner, y_i^{SEL}: $D_i = y_i^{\mathrm{TOP}} - y_i^{\mathrm{SEL}}$. As noted in Chapter 4, the error D_i can also be interpreted as a distance to the "virtual best" performance for a given dataset, x_i.

Using the identified meta-features to generate a multidimensional feature space populated by datasets, the local variation of D_i can be quantified using the n nearest neighbours of the dataset, x_i, as $s(D_l)_i$, where $l = 1, \ldots, n$ and $s(D_l)$ denotes the standard deviation of D_l.

The literature examining the reliability and uncertainty of individual predictions of machine learning models, in particular of regression models (Shah et al. 2022; Zaoui et al. 2020), proposed the use of a *conditional variance*, which represents the variance of the model's residuals, given a certain input. Inspired by this approach, the $\textsc{Risk}_{x_i}$ for the dataset x_i is defined as a local variation $s(D_l)_i$ of the meta-learner's error D_i:

$$\textsc{Risk}_{x_i} = s(D_l)_i \tag{5.2}$$

To assess the $\textsc{Risk}_{x_i'}$ for a new dataset, x_i', it is proposed to identify a region defined by this new dataset's neighbourhood, comprised of the regions where original datasets x_i are located. Given the nature of the measure, such that the risk represents the potential worst-case scenario for a specific location, the $\textsc{Risk}_{x_i'}$ for a new dataset is defined as the maximal $\textsc{Risk}_{x_i}$ measure of the new dataset's neighbourhood:

$$\textsc{Risk}_{x_i'} = \max\left\{\textsc{Risk}_{x_i}^{l} : l = 1, \ldots, n'\right\}, \tag{5.3}$$

where n' denotes the number of the nearest neighbours of the dataset x_i' from the original set of datasets x_i.

The above enables the first question posed in this chapter to be addressed. The terminology required to tackle the second question is defined in the following part of this section.

Consider the probability PC that the meta-learner's error D_i would not exceed a certain value, denoted as D_{PC}. This error value is later referred to as the *upper-bound error* for a given probability PC. According to the second question formulated in this chapter, the goal is to assess the upper-bound error D_{PC} for the accepted confidence level (probability) PC, and for the Risk value estimated for a specific dataset.

It is assumed that the probability distribution of errors D_i is a function of Risk:

$$P(D_i) = h\left(\textsc{Risk}_{x_i}\right) \tag{5.4}$$

where h denotes an experimentally obtained mapping function between Risk values and $P(D_i)$. Estimating the mapping h enables a determination of the upper-bound errors D_{PC} for a given Risk:

$$D_{\text{PC}} = h^*\left(\textsc{Risk}, \text{PC}\right), \tag{5.5}$$

where h^* denotes a function mapping derived from h. Therefore for a new dataset x_i' and the estimated value of $\textsc{Risk}_{x_i'}$ (Equation 5.3), the upper-bound error can be established using the same mapping h^*:

$$D_{\text{PC}}\,|_{x_i'} = h^*\left(\textsc{Risk}_{x_i'}, \text{PC}\right), \tag{5.6}$$

The above addresses the second question of the formulated problem. The methodology employed to estimate both the Risk and the upper-bound error D_{PC} is presented in the following section.

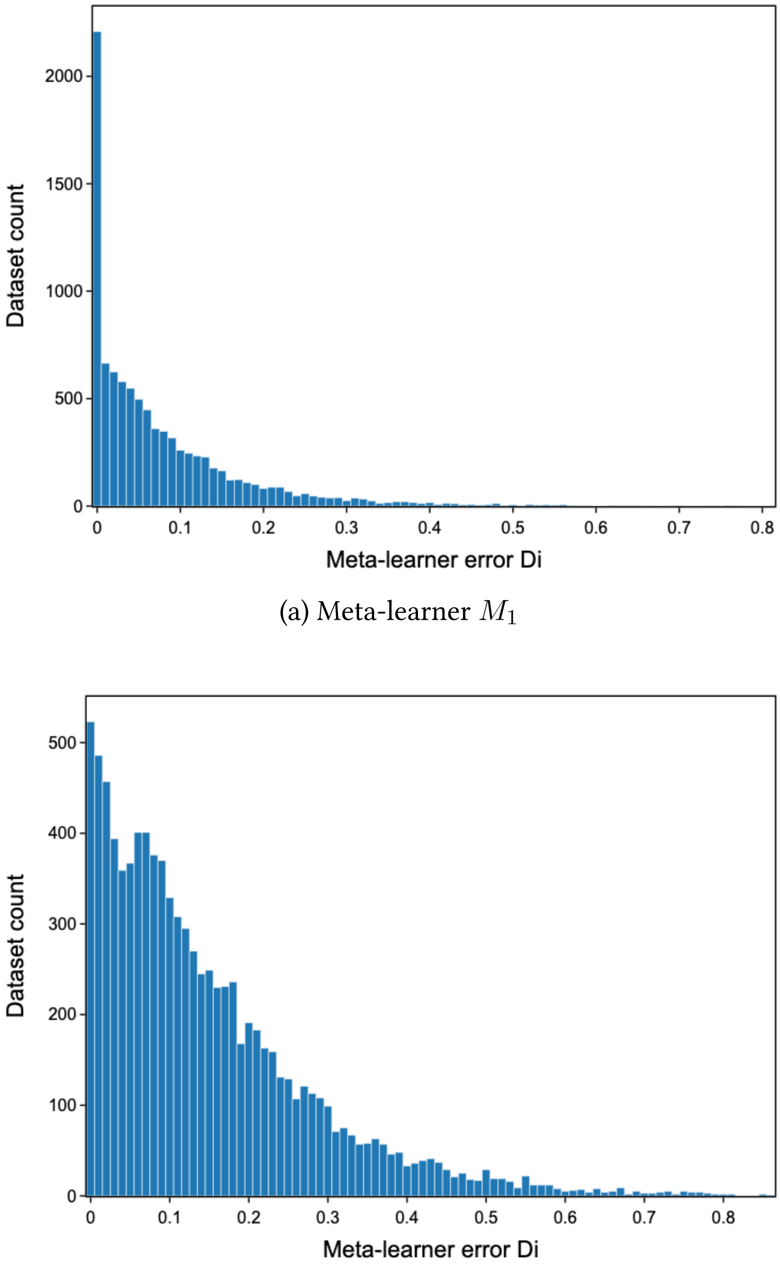
5.2 Methodology

In the present study, two variants of the meta-learner, denoted as $M_1 := M(m_1, F_1, A_1)$ and $M_2 := M(m_1, F_1, A_2)$, were investigated. The components m_p, F_q, and A_r are elucidated in Table 4.3 of Chapter 4. The choice to employ feature set F_1 was pragmatically motivated by a preference for a lower-dimensional space as opposed to a sparser, high-dimensional space. This decision was further justified by the observation that the performance of the meta-learner was relatively invariant to the particular feature set utilised. The NN-based meta-models were selected over the alternative approach due to their superior performance and computational efficiency during training, which enabled the execution of multiple experiments. The AUC metric was employed in this experiment to produce meta-learners' performance measures. This metric was preferred over the AP because it is more appropriate for comparing datasets together and has a less skewed distribution across all datasets.

Procedure 6 Processing original set of datasets

```
Input:
    $F \in \mathbb{R}^{N\times K}$, $Y_{\text{AUC}} \in \mathbb{R}^{N\times L}$, where:
      $N$ – number of datasets,
      $K$ – number of meta-features,
      $L$ – number of base AD algorithms
Output:
    Meta-learner errors $\{D_i \, : \, i = 1, \dots, N\}$
Create six complementary subsets from $N$ datasets
for iteration it $= 1$ to $6$ do
    { Use 5:1 subset split, with test corresponding to it }
    $(F_{\text{it}}^{\text{train}}, F_{\text{it}}^{\text{test}}) \leftarrow F$
    $(Y_{\text{it}}^{\text{train}}, Y_{\text{it}}^{\text{test}}) \leftarrow Y$
    $m_{\text{it}} \leftarrow \text{train}(F_{\text{it}}^{\text{train}}, Y_{\text{it}}^{\text{train}})$ { Meta-model supervised training }
    $\widehat{Y}_{\text{it}}^{\text{test}} \leftarrow m_{\text{it}}(F_{\text{it}}^{\text{test}})$ { Predict }
end for
$\widehat{Y} \leftarrow \text{concatenate}\left\{\widehat{Y}_{\text{it}}^{\text{test}}\right\}$
Obtain $y_i^{\text{SEL}}$ with Eq. 4.10
Obtain $D_i$ with Eq. 4.11
return $\{D_i\}$
```

To acquire the errors D_i over a wide range of datasets, the meta-models were trained in a 6-fold manner, with a model trained on five dataset folds for each iteration, and the predictions, along with the errors D_i, obtained using the sixth fold. Procedure 6 illustrates the steps taken to obtain the meta-learners' errors.



(a) Meta-learner M_1

(b) Meta-learner M_2

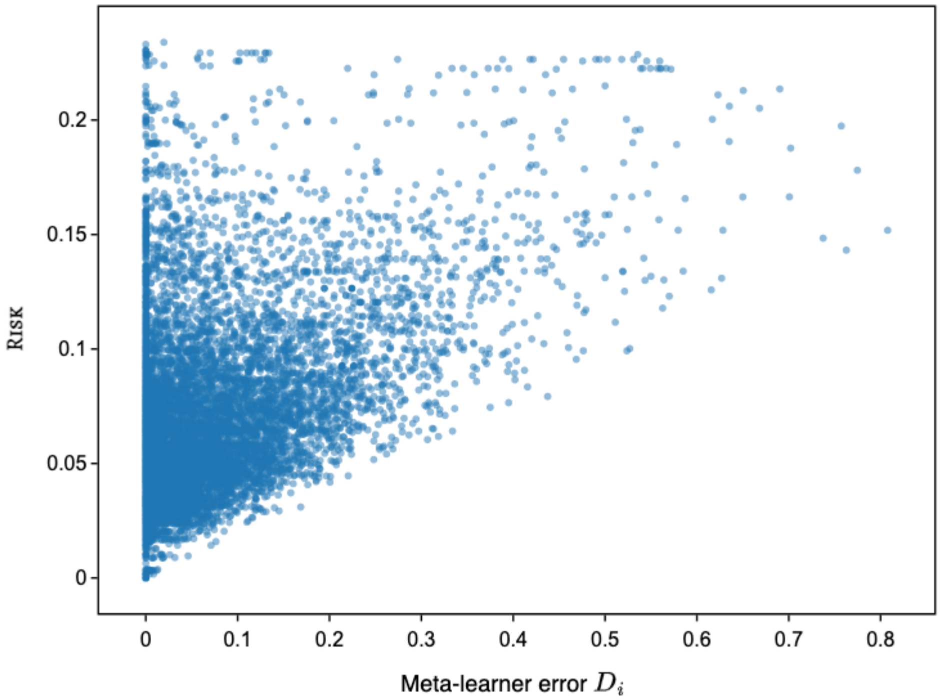
Figure 5.1: Histogram of meta-learner errors D_i.

Figure 5.1 shows the distribution of errors obtained across datasets x_i. The error counts for the meta-learner M_1, which was trained on 13 base AD algorithms, show a very steep decline for error values slightly greater than zero. This is a highly desirable behaviour because it demonstrates that the meta-model selects the best possible solution for a large proportion of datasets. The second meta-learner, M_2, which used 298 algorithms, exhibits a much slower decrease in the number of errors as their value rises. The difference is not surprising, given that the second meta-learner had a larger collection to choose from. It is worth noting that, while its efficiency appears to be lower, the average performance of this meta-learner was still higher than the M_1's performance.

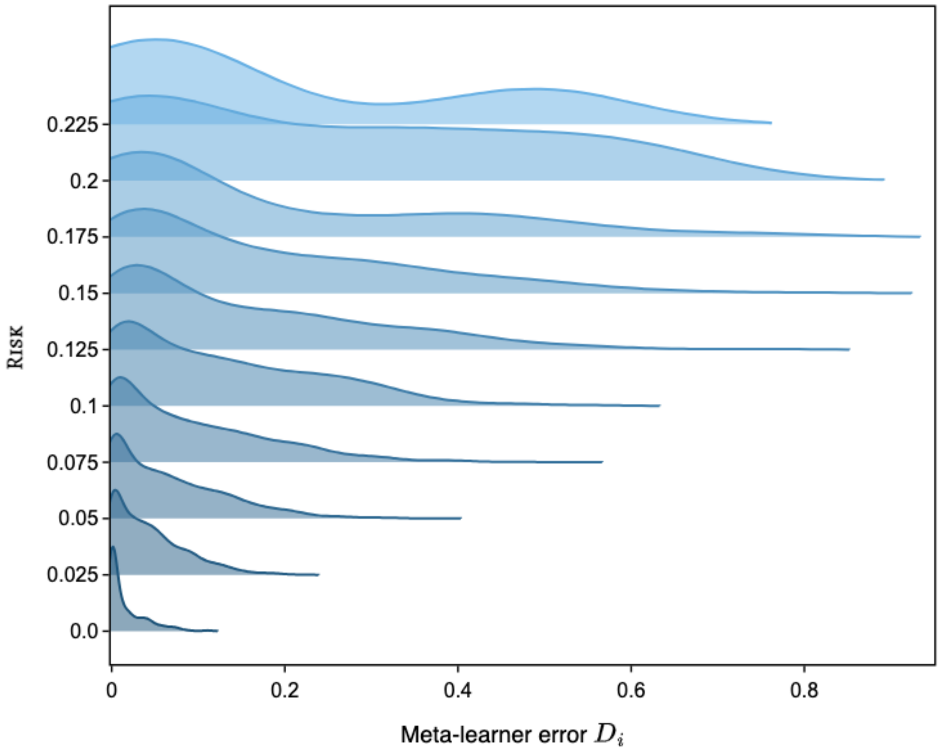
Neighbourhood areas of $n = 30$ datasets have been formed around each dataset, x_i, by measuring the Euclidean distance on the space of scaled meta-features. Scaling was performed using the mean and standard deviation for normalisation. The size of n was chosen to strike a balance between ensuring enough samples in each neighbourhood and reducing its size to allow for local fluctuations to be captured. The standard deviations of error D_i were calculated for each neighbourhood. The $\textsc{Risk}_{x_i}$ around the dataset x_i has been computed as the standard deviation of the errors from n nearest neighbours of x_i, as shown in Equation 5.2.

The arrangement of datasets according to their error D_i and their $\textsc{Risk}_{x_i}$ values, for one of the meta-learners, M_1, is shown in Figure 5.2a. Another perspective on the same data, which closer illustrates the Equation 5.4, is presented in Figure 5.2b. In the observed data, errors of lower magnitude are dispersed across the entire spectrum of considered Risk levels. This is advantageous as it implies that the meta-learner is capable of generating accurate predictions even in more unstable regions characterised by higher Risk. Importantly, the data also reveals an absence of large errors in the lower-risk regions. While this result is consistent with expectations, it serves as an empirical base to formulate the hypothesis that the likelihood of encountering high-magnitude errors increases as a function of Risk.

When interpreting values of errors D_i, it is essential to note that the scale of this error metric is comparable to that of the AUC metric – it represents the difference between two AUC values. Given that the range of possible AUC values lies between 0 and 1, and taking into account that D_i is non-negative by definition, it follows that the range of D_i is also [0, 1]. Nonetheless, it should be highlighted that, in practice, reasonable AUC values typically fall within the [0.5, 1] interval. Consequently, any D_i value exceeding



(a) Arrangements of individual datasets x_i.



(b) Variable distributions $P(D_i)$ versus increasing risk levels.

Figure 5.2: Distribution of the datasets according to their D_i and RISK_{x_i}.

0.2 may be considered as representing a medium-to-large error magnitude.

Assuming that shapes of the distribution $P(D_i)_{\textsc{Risk}_i}$ for any $\textsc{Risk}_{x_i}$ are similar and proportionally stretched to the $\textsc{Risk}$ value, the mapping function h was formulated as outlined in Equation 5.4. This was achieved by constructing lines of the form: $\textsc{Risk} = a_{\text{PC}}D$ for a predefined set of coefficient values a_{PC}, followed by computing the proportion of the datasets situated above each corresponding line. Hence, a mapping between percentage values and the coefficients $a_{\text{PC}} = \textsc{Risk}_i/D_i$ was established. This mapping was then interpolated to produce the relationship φ depicted in Figure 5.3. The procedural steps are outlined in Procedure 7. The derived relationship serves as a basis for evaluating the upper-bound error D_{PC} for new datasets x'_i, for which $\textsc{Risk}_{x'_i}$, had been previously estimated.

Procedure 7 Creation of mapping h^*

```
Input:
    Predefined coefficients $\{a_{\text{PC}}\}$
    $D_i$, $\textsc{Risk}_i$, where $i = 1, \ldots, N$
Output:
    Mapping $h^*$
for all $a_{\text{PC}}$ do
    count $= 0$
    for $i = 1$ to $N$ do
        if $\textsc{Risk}_i > a_{\text{PC}}D_i$ then
            count $++$ { Count all the datasets above the line }
        end if
    end for
    PC $\leftarrow$ count$/N$ { Calculate the datasets proportion }
end for
$\varphi(\text{PC}) = a \leftarrow$ interpolate $(\text{PC} \mapsto a_{\text{PC}})$
$h^*(\textsc{Risk}, \text{PC}) \leftarrow \textsc{Risk}/\varphi(\text{PC})$
return $h^*$
```

To evaluate the proposed methodology, a validation set comprising 1,000 datasets, initially withheld during the experimental setup, was utilised. Preliminary procedures, including the generation of meta-features, predictions with the meta-learner, and the

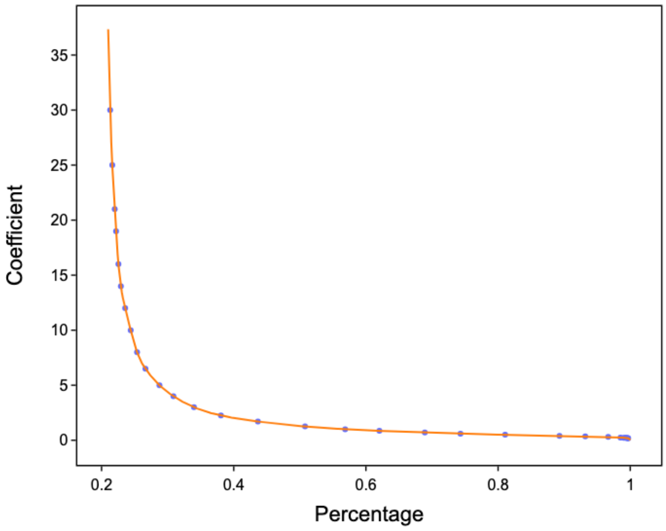


Figure 5.3: The interpolated relationship φ between the percentage PC and the coefficient a_{PC}.

computation of the D_i error metrics, were executed on this validation set. For the normalisation of meta-features, the scaling transformation developed on the original set of datasets was applied. For each dataset x'_i in the validation set, the risk level $\textsc{Risk}_{x'_i}$ was determined by identifying $n' = 10$ closest neighbours from the original set and applying the maximal value, as delineated by Equation 5.3. The rationale behind selecting $n' = 10$ is elaborated upon in Section 5.3. Further, based on the relationship h^*, upper-bound errors $D_{0.5}$ and $D_{0.95}$, representing the 50th and 95th percentiles, for arbitrarily chosen percentiles, were calculated for each dataset x'_i, as specified in Equation 5.6.

The findings from the above analysis, along with an interpretation and discussion, are presented in the following section.

5.3 Results and Discussion

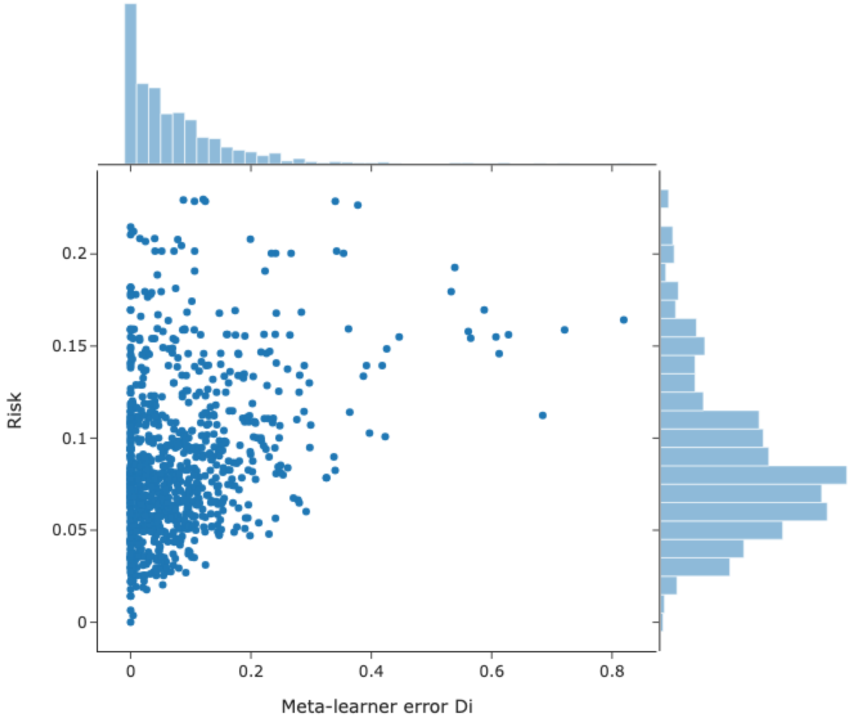
To visualise the relation between the assigned $\textsc{Risk}_{x'_i}$ and the actual meta-learner errors, D_i, the validation datasets, x'_i, were organised as illustrated in Figure 5.4, for both meta-learners, M_1 and M_2. The figure additionally shows the histograms of both characteristics for the validation set. An initial observation is that the bottom-right quadrants of both images contain no or a minimal number of datasets, given the allocated Risk levels. This observation is important as it provides encouragement for the hypothesis that datasets located in low-risk regions are less prone to exhibiting high error values.

To strengthen the above illustration, a cross-check was conducted between the assigned levels of $\textsc{Risk}_{x_i'}$ and the actual errors D_i within the validation set of datasets x_i'. For ease of interpretation, the assigned risks were segmented into three categories: *low*, *medium*, and *high*. The demarcation thresholds were established by splitting the $\textsc{Risk}$ range into three equal intervals, and they were chosen for illustration only. Figure 5.5 provides a box-plot representation of the actual errors, classified by the specified risk level. The figure incorporates data processed with both meta-learners M_1 and M_2.

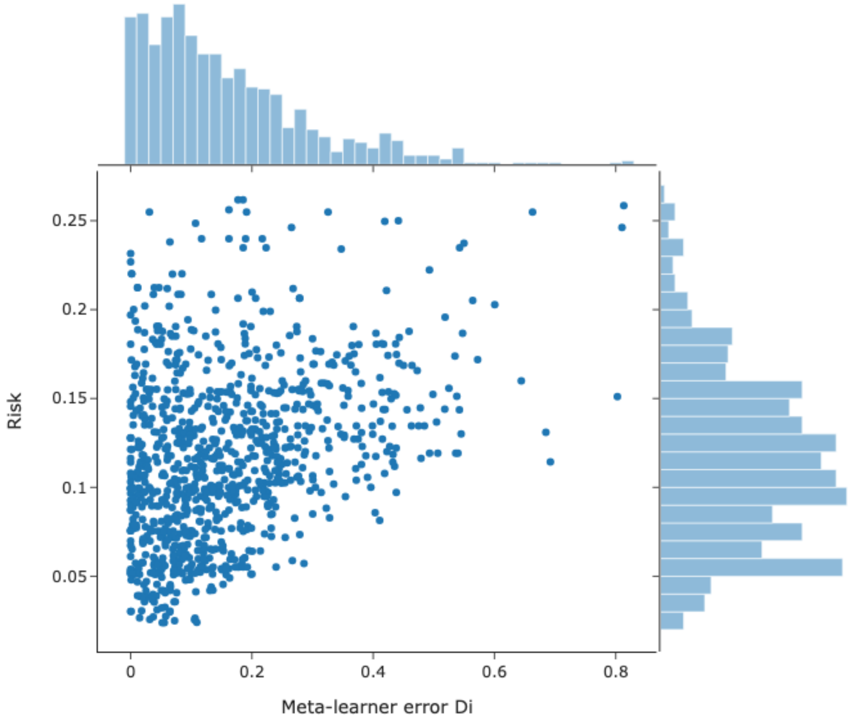
The analysis reveals that across both meta-learners, the median error, the upper fence, and the spread of errors exhibit an increasing trend from the low-risk to the high-risk categories. While numerous datasets with low errors D_i have fallen into the high-risk category, it is noteworthy that the maximum error for datasets in the low-risk category does not exceed 0.3. Furthermore, datasets with the highest errors predominantly fall into the high-risk category, apart from a few outliers categorised as medium-risk – particularly in the case of the M_2 meta-learner. These findings are significant as they lend empirical support to the hypothesis that datasets situated in regions categorised as low-risk are less likely to have greater error values. While it is acceptable for datasets in high-risk areas to manifest lower errors, it is essential that datasets in low-risk regions do not exhibit high error values.

The results of estimating upper-bound errors D_{PC} within a given probability range are presented in the following part of this section.

For each validation dataset x_i', the values of D_{PC} have been calculated using the assigned Risk$_{x_i'}$, for two percentiles: 0.95 and 0.5. Figure 5.6 shows the charts with the datasets organised according to their corresponding actual errors D_i and the associated $\textsc{Risk}$ levels, with colours representing the probability ranges. Specifically, red markers correspond to instances where the error exceeds the 95th percentile, occurring with a probability less than 0.05 for any given $\textsc{Risk}$. Orange markers signify instances where the error is greater than the median but remains below the 95th percentile across all

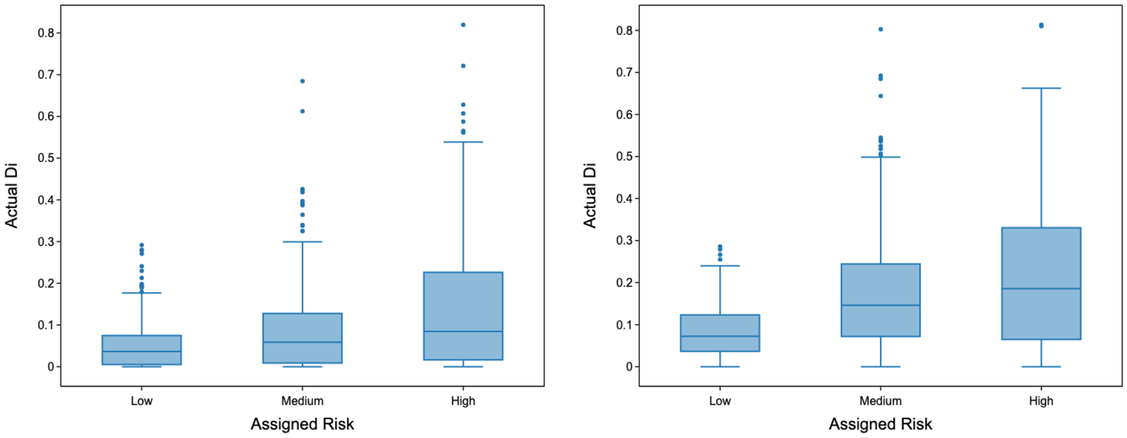


(a) Meta-learner M_1



(b) Meta-learner M_2

Figure 5.4: Validation datasets distributed according to their actual D_i and assigned $\textsc{Risk}_{x_i'}$.



(a) Meta-learner M_1

(b) Meta-learner M_2

Figure 5.5: The distributions of the actual errors D_i within low, medium, and high categories of the assigned Risk for validation datasets.

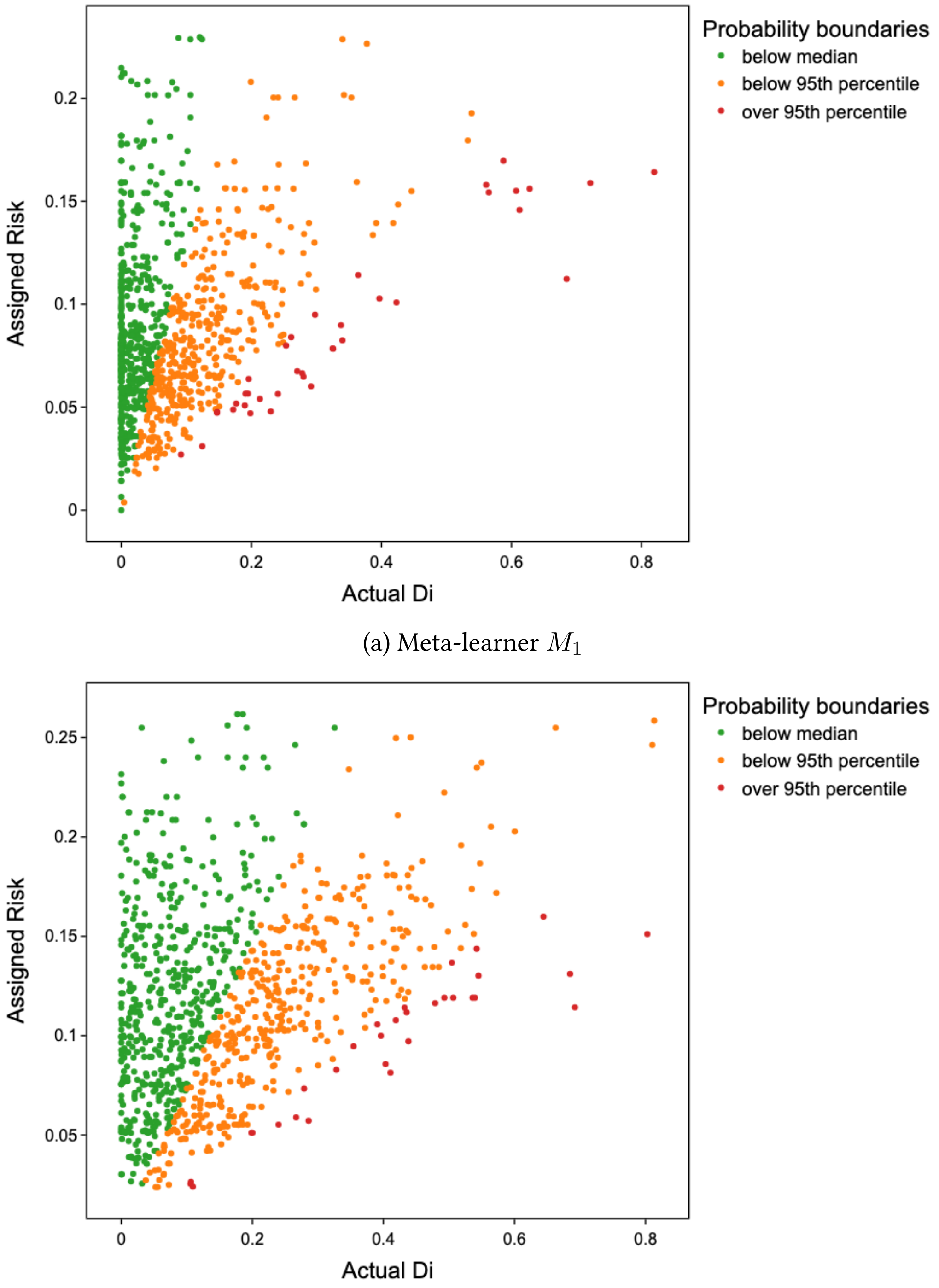
Table 5.1: Proportions of datasets with errors below the upper-bound error D_{PC} for two meta-learners.

	M_1	M_2
$D_i \leqslant D_{0.5}$	0.551	0.548
$D_i \leqslant D_{0.95}$	0.962	0.969

risk levels. Similarly, green markers represent cases where the error does not surpass the median. The layout of the validation datasets, consistent with the box-plot analysis, reveals that low-risk zones are free of occurrences with high errors.

To verify the obtained upper-bound errors D_{PC}, the proportion of datasets with errors D_i below the $D_{0.5}$ and $D_{0.95}$ were calculated. Table 5.1 presents the proportions for both investigated meta-learners, M_1 and M_2. The results for both meta-learners demonstrate that the fraction of datasets associated with errors smaller than D_{PC} surpassed the estimated proportion. This suggests that the actual meta-learners' performance was relatively good more frequently than the expectations set by the initial probability thresholds. Consequently, the data provides evidence that the estimated probability boundaries possess predictive potential. New datasets tend to conform to similar distributions and reside within the probabilistic ranges established with the proposed procedure.

Eventually, an assessment was conducted to ascertain the influence of the neighbourhood size parameter n' on the results. The value of n' inherently impacts the assigned Risk metric and, as a result, the error boundary estimates. Upon varying n' within



(a) Meta-learner M_1

(b) Meta-learner M_2

Figure 5.6: Validation datasets according to their error probability boundaries.

the range of $1, \ldots, 13$, the data suggested that its impact on the proportion of datasets within the examined range appeared to be stochastic. However, the parameter exhibited a pronounced influence on the outlier datasets. Specifically, for $n' \leqslant 5$, there was a noticeable migration of outlier datasets characterised by low Risk and elevated errors D_i to the lower-right quadrant of the graphical representation. The presence of datasets in this quadrant is highly undesirable for the intended outcome of this study. Based on the analysis, an optimal value of $n' = 10$ was identified as the most suitable for both meta-learners.

The above analysis demonstrates that for a given meta-learner, it is feasible to identify regions of instability indicating decreased reliability of the meta-learner's predictions. Additionally, this study provides the framework for quantifying the risk of an inaccurate prediction across the meta-feature space. Consequently, this facilitates an assessment concerning the reliability of the meta-learner's predictions on previously unseen and unlabeled datasets.

The analysis also demonstrates that it is possible to estimate the probability boundaries for meta-learner errors. In practical terms, this offers a statistical basis for approximating both the frequency and magnitude of errors the meta-learner is likely to commit within different regions of the meta-feature space. When applying the estimated probability boundaries to a novel problem, the errors made by the meta-learner are likely to fall within the estimated range with the expected frequency.

The observation that regions of instability can be identified during meta-learning examinations revives the discussion around the "No Free Lunch" theorem, which initially inspired this study. This theorem appears to extend to the level of meta-learning, suggesting that an ideal meta-learner capable of solving all issues with high accuracy may not exist.

In light of this theorem, it is recommended to enhance the construction of a meta-learner, particularly in the context of unsupervised AD, by incorporating information about regions prone to high error rates. This additional layer of information could serve as a valuable tool for practitioners to gauge the reliability of specific predictions tailored to their individual problems. This is especially pertinent when taking into account the relatively early stages of research in the domains of algorithm selection in unsupervised settings and local reliability estimates.

This section addressed the research question *RQ3*, which focuses on the identification of high-risk regions within the input space and the assessment of the reliability of individual meta-learner responses. Additionally, it responds to the research question *RQ4*, which examines the impact of the dataset size on meta-learner analysis. The outcomes of this study heavily rely on the large volume of datasets utilised in the analysis. The substantial dataset volume is critical, as it enables the derivation of statistical estimates and the validation of formulated hypotheses. Without such volume, neither statistical inference nor their verification would be feasible.

Chapter 6

Conclusions

In this work, the meta-learning approach to algorithm selection for unsupervised anomaly detection (AD) was explored. Two aspects highlight the importance of this research area. The first is the escalating significance of anomaly detection techniques due to the significant increase in data gathered from a diverse array of devices used in various activities of daily life. The second is an absence of systematic methods for tackling unsupervised AD tasks, including techniques for automatically selecting the appropriate algorithm for a given task.

The current research was carried out by considering the following research problems:

1. Development of a meta-learner suited to unsupervised AD.
2. Examination of the impact of individual meta-learner components (meta-model, meta-features and base set of algorithms) on its overall performance.
3. Providing a strategy for assessing the reliability of particular meta-learner predictions.
4. Conducting experiments using a comprehensive collection of benchmark datasets, which represents the largest dataset compilation to date in the field of algorithm selection for unsupervised AD.

The above aspects have been encapsulated in the respective research questions outlined in this study. The subsequent sections of this chapter provide a structured conclusion and future outlook. Specifically, Section 6.1 concludes the experimental work conducted and the results obtained from each study. Following this, Section 6.2 identifies limitations in the current work and recommends future research directions to address these challenges.

6.1 Contributions of this Thesis

6.1.1 Meta-Learner for Unsupervised AD

A novel meta-learner was introduced, designed to facilitate the selection of an optimal unsupervised algorithm for AD tasks. Compared to the existing state-of-the-art solution, MetaOD (Zhao, Rossi, et al. 2021), a statistically significant improvement was observed in the proposed method based on two critical performance metrics: Area Under the Receiver Operating Characteristic Curve (AUC) and Average Precision (AP). Additionally, substantial savings in training time were achieved by the newly presented meta-learner in comparison to the matrix-factorisation-based MetaOD. While the reduced time frame for training has no impact on the prediction phase, it holds implications for the offline training stage, as well as for potential redesign, retraining, or fine-tuning with additional data.

Overall, this study demonstrates that the benefits of integrating a meta-learner into an AD pipeline could easily outweigh the associated costs. Assuming the dataset size to be 1,000 observations and 45 features, adding an extra 1–2 seconds on meta-feature generation and 0.5 seconds on identifying the appropriate algorithm will potentially save hours of iterative or exhaustive testing and assessing of algorithm candidates.

6.1.2 Contribution of Meta-Learner Components

In the second problem, a 2^3 experimental design was employed to investigate the individual impact of component parts – namely the meta-model, meta-features, and base algorithm set – on the overall performance of the meta-learner in both the MetaOD and the proposed approach. The findings indicate that while the selection of meta-features and the base set of AD algorithms exerted a relatively minor influence, the choice of meta-model emerged as the most significant factor affecting meta-learner performance.

Moreover, the analysis unveiled that a hybrid approach, incorporating elements from both MetaOD and the method proposed in this study, yielded the most favourable performance outcomes.

The current emphasis in AutoML and meta-learning literature is primarily on meta-feature generation and hyperparameter optimisation. However, the current research suggests that investing in significant pre-evaluation of a wide range of algorithms via comprehensive grid search approaches delivers only modest benefits in terms of feasible algorithms and hyperparameters. Instead, this study argues that the most efficient way to create effective meta-learners in the domain of unsupervised AD is to concentrate on designing a meta-model that can efficiently leverage data from previous evaluations.

The work on the development of the meta-learner and its characteristics has been published in *IEEE Access* (Gutowska et al. 2023).

6.1.3 Local Reliability and Risk

Further to the development of the meta-learner, a strategy for evaluating the reliability of individual meta-learner predictions was introduced. This framework is based on analysing the meta-feature space to locate areas with a greater likelihood of generating unreliable predictions. It also delivers probabilistic estimates indicating the likelihood that the suggested algorithm would diverge from the optimal selection by a certain amount in terms of the performance metric used. Empirical results demonstrate the framework's capabilities in providing the upper bound on the potential error.

This study emphasises the necessity of equipping a meta-learner with additional meta-data on the reliability of individual predictions when constructing such a meta-learner. The need for such contextual data becomes particularly relevant in algorithm selection for unsupervised AD for several reasons. First, the unsupervised nature of the problem limits an immediate validation of the algorithm's performance, making a reliability assessment tool invaluable. Second, the feature space in AD tasks often exhibits high non-linearity, making some tasks inherently more challenging for the meta-learner to accurately address. Finally, meta-learning is natively case-specific; each observation in the meta-feature space corresponds to a separate AD problem with unique characteristics, therefore, a general measure of efficiency is not particularly informative.

By providing insights into the potential risk of inaccuracy for specific cases, the framework empowers practitioners to make informed decisions. They can gauge the criticality of the task at hand and assess the risk associated with the algorithm's potential error, thus tailoring their approach accordingly.

6.1.4 Dataset Collection

The experimental work in this study was conducted using the most extensive collection of datasets used to date in meta-learning experiments focused on unsupervised AD. This substantial size enabled some unique accomplishments that would not be possible with smaller dataset collections. These include:

- Establishing a well-populated meta-feature space,
- Conducting a thorough comparison between the proposed meta-learner and the existing state-of-the-art solution,
- Identifying new insights into how different components of the meta-learner contribute to its overall performance,
- Analysing the meta-feature space to pinpoint high-risk areas and provide estimates of the meta-learner's error rates.

In summary, the expansive scale of this study significantly enhanced the capacity to uncover novel insights, exceeding what is typically achievable with the more limited dataset sizes used in other meta-learning research.

6.2 Future Work

6.2.1 Development of Robust Datasets

In the evolving landscape of meta-learning systems for unsupervised AD, there is a pressing need for the development of dataset benchmarks that serve as a truly robust test for these systems. The inclusion of cross-domain data, data of diverse types, and semantically meaningful data, along with sophisticated diversity measures, should all be at the core of future directions.

Although the dataset collection used in this work is large, it was derived from a relatively narrow base set that was augmented using iterative sampling techniques. While this methodology is effective at producing large numbers of datasets, it might restrict their diversity, limiting meta-learner adaptability and generalisability. To overcome this limitation, future research should prioritise the creation of datasets that are inherently diverse and drawn from a variety of domains. Extending on this theme, meta-learning research will benefit from datasets derived from a variety of data types, including time series data.

A truly robust dataset also needs to address specific real-world problems typical for anomaly detection. The dataset collection used in this study is missing datasets with semantically meaningful attributes, potentially making meta-learners less effective in practical applications.

The pursuit of diversity, however, does not end with mere inclusion. The metrics used to assess this diversity must be tailored to specific needs. These could be built upon entropy-based metrics or topological features to capture the nuanced characteristics required for a rigorous test of meta-learner capabilities for anomaly detection.

6.2.2 Finding a Threshold Between Normal and Anomalous Data

Anomaly detection techniques do not inherently offer the ability to establish a boundary between normal and anomalous data but instead, compute such a boundary based on a required input parameter describing the proportion of anomalies in a given dataset. Threshold-independent metrics like AUC and AP provide valuable insights into the effectiveness of AD algorithms, but similarly, they are limited in assisting in the establishment of such a boundary.

The combination of original dataset features and associated meta-features could offer an innovative approach to this boundary specification problem. Meta-features tailored to anomaly detection needs can encapsulate useful underlying patterns of the data. By incorporating meta-features, a richer feature space is created, allowing for a more nuanced analysis of the boundary conditions between normal and anomalous points.

This opens up an intriguing avenue for future meta-learner development: the design of specialised systems aimed not only at optimising algorithm selection, but also at defining the intricate boundary separating normal data from anomalies. Such research would help to improve the effectiveness of existing anomaly detection methods as well as provide a more comprehensive understanding of the fundamental characteristics that define anomalous behaviour in complex datasets.

6.2.3 Boundary Parameter Refinement

In the analysis of the reliability and risk of individual meta-learner responses, the Risk metric was introduced. The metric values were categorised into three buckets – low, medium, and high. While the categorisation has provided a structured framework for evaluating the risk of inaccurate predictions, it has merely served as an initial concept for illustration rather than a sophisticated representation.

The current division fails to offer detailed insights that could be invaluable for practitioners implementing the Risk metric in real-world settings. As a result, another potential future path would be to delve deeper into defining these categories, ensuring that each category bears substantive meaning for those applying the tool.

Additionally, a one-size-fits-all approach to risk categorisation is unlikely to serve the variety of needs present in diverse domains. Different sectors – finance, healthcare, or manufacturing – may have unique requirements that a universal classification could not properly encapsulate. Consequently, any future work dedicated to refining boundary parameters should consider adapting the risk categories based on domain-specific characteristics. Such an approach would result in a more responsive and insightful metric, offering better utility across a broader range of applications.

6.2.4 Explaining Meta-Learners

One of the most fundamental observations made during this research was the disparity in performance between different AD algorithms when applied to identical AD problems. This observation highlights the importance of an in-depth understanding of the mechanisms underlying these algorithms in order to facilitate their effective deployment in a variety of scenarios.

While the use of meta-learners has shown promise in closing performance gaps by selecting optimal algorithms based on dataset features, these meta-learners are not without limitations. Typical of complex machine learning methods, they lack the ability to provide transparent rationales for their decisions. This is, however, a significant disadvantage because it makes the meta-learner's recommendations difficult to interpret, limiting trust and possible adoption of these tools in real-world applications.

Another avenue for future work could therefore be the exploration of the meta-learner's decision-making processes, potentially through examining the complex landscape of its loss function. The additional quantity that can be insightful in such analysis is the meta-learner error, D_i, introduced in Chapter 4 and leveraged in both Chapters 4 and 5. This error represents the distance between the recommended and the "virtual best" algorithm. Deeper insight into the meta-learner decision-making process can be obtained by examining the factors driving these errors.

By gaining these insights, it could be possible to create a set of human-interpretable rules. Such a rule set would serve dual purposes. First, it could act as a substitute for a meta-learner in scenarios where deploying such advanced tools is not feasible. Secondly, it could serve as a verification mechanism to validate the meta-learner's choices, providing reassurance that the suggestions are based on logical and empirically supported criteria.

6.3 Final Summary

The current research work presents substantial advancements in the domain of meta-learning, specifically focusing on its application to algorithm selection in unsupervised anomaly detection (AD). The study offers the following primary contributions: the creation of the meta-learner designed for unsupervised AD tasks, the demonstration that the architecture and design of the meta-model are crucial for effective meta-learning, and the introduction of a framework for evaluating errors a meta-learner may commit in its individual predictions. Moreover, this investigation employs a comprehensive collection of datasets, providing a robust foundation for its findings.

The work addresses a significant gap in the existing literature by offering a systematic methodology for algorithm selection in unsupervised anomaly detection. This is particularly pertinent given the rapid expansion of data volumes and the corresponding need for effective anomaly detection mechanisms. The research thus not only contributes to the academic discourse but it provides a comprehensive and actionable set of tools for both researchers and practitioners. As such, this work facilitates more effective data management in an increasingly data-rich world.

Bibliography

Abdrashitova, Yulia, Alexey Zabashta, and Andrey Filchenkov (2018). “Spanning of Meta-Feature Space for Travelling Salesman Problem”. In: *Procedia Computer Science* 136. 7th International Young Scientists Conference on Computational Science, YSC2018, 02-06 July2018, Heraklion, Greece, pp. 174–182. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.08.250 (cit. on p. 32).

Ahmed, Faruk and Aaron Courville (2020a). “Detecting semantic anomalies”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04, pp. 3154–3162 (cit. on p. 23).

— (2020b). “Detecting semantic anomalies”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04, pp. 3154–3162 (cit. on p. 48).

Ali, Rahman, Sungyoung Lee, and Tae Choong Chung (2017). “Accurate multi-criteria decision making methodology for recommending machine learning algorithm”. In: *Expert Systems with Applications* 71, pp. 257–278 (cit. on p. 15).

An, Jinwon and Sungzoon Cho (2015). “Variational autoencoder based anomaly detection using reconstruction probability”. In: *Special Lecture on IE* 2.1, pp. 1–18 (cit. on p. 26).

Bahri, Maroua, Flavia Salutari, Andrian Putina, and Mauro Sozio (2022). “AutoML: state of the art with a focus on anomaly detection, challenges, and research directions”. In: *International Journal of Data Science and Analytics* 14.2, pp. 113–126 (cit. on p. 32).

Bahrpeyma, Fouad, Mark Roantree, Paolo Cappellari, Michael Scriney, and Andrew McCarren (2021). “A methodology for validating diversity in synthetic time series generation”. In: *MethodsX* 8, p. 101459 (cit. on p. 44).

Bauder, Richard A. and Taghi M. Khoshgoftaar (2017). "Estimating outlier score probabilities". In: *2017 IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE, pp. 559–568 (cit. on p. 46).

Biewald, Lukas (2020). *Experiment Tracking with Weights and Biases.* Software available from wandb.com (cit. on pp. 61, 69).

Bischl, Bernd, Pascal Kerschke, Lars Kotthoff, Marius Lindauer, Yuri Malitsky, Alexandre Fréchette, Holger Hoos, Frank Hutter, Kevin Leyton-Brown, Kevin Tierney, et al. (2016). "Aslib: A benchmark library for algorithm selection". In: *Artificial Intelligence* 237, pp. 41–58 (cit. on p. 15).

Bosnić, Zoran and Igor Kononenko (2008a). "Comparison of approaches for estimating reliability of individual regression predictions". In: *Data & Knowledge Engineering* 67.3, pp. 504–516 (cit. on p. 38).

— (2008b). "Estimation of individual prediction reliability using the local sensitivity analysis". In: *Applied intelligence* 29, pp. 187–203 (cit. on p. 38).

— (2009). "An overview of advances in reliability estimation of individual predictions in machine learning". In: *Intelligent Data Analysis* 13.2, pp. 385–401 (cit. on p. 38).

— (2010). "Correction of regression predictions using the secondary learner on the sensitivity analysis outputs". In: *Computing and Informatics* 29.6, pp. 929–946 (cit. on p. 38).

Breunig, Markus M., Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander (2000). "LOF: identifying density-based local outliers". In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104 (cit. on pp. 23, 24, 27, 51, 59).

Bulusu, Saikiran, Bhavya Kailkhura, Bo Li, Pramod K. Varshney, and Dawn Song (2020). "Anomalous example detection in deep learning: A survey". In: *IEEE Access* 8, pp. 132330–132347 (cit. on pp. 22, 23).

Campos, Guilherme O, Arthur Zimek, Jörg Sander, Ricardo JGB Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E Houle (2016). "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study". In:

Data mining and knowledge discovery 30, pp. 891–927 (cit. on pp. 14–17, 28–30, 35, 36, 41–43, 47, 49, 59).

Chalapathy, Raghavendra and Sanjay Chawla (2019). "Deep Learning for Anomaly Detection: A Survey". In: *CoRR* abs/1901.03407 (cit. on pp. 14, 22).

Chandola, Varun, Arindam Banerjee, and Vipin Kumar (July 2009). "Anomaly Detection: A Survey". In: *ACM Computing Surveys (CSUR)* 41.3. ISSN: 0360-0300. DOI: 10.1145/1541880.1541882 (cit. on pp. 14, 22, 27, 28).

Chollet, François et al. (2015). *Keras*. https://keras.io (cit. on p. 68).

Cohen, Jacob (2013). *Statistical power analysis for the behavioral sciences*. Academic press (cit. on pp. 63, 64).

Davis, Jesse and Mark Goadrich (2006). "The relationship between Precision-Recall and ROC curves". In: *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240 (cit. on p. 48).

Demidenko, Eugene (2013). *Mixed models: theory and applications with R*. John Wiley & Sons (cit. on p. 70).

Di Mattia, Federico, Paolo Galeone, Michele De Simoni, and Emanuele Ghelfi (2019). "A survey on gans for anomaly detection". In: *arXiv preprint arXiv:1906.11632* (cit. on p. 26).

Emmott, Andrew, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong (2013). "Systematic construction of anomaly detection benchmarks from real data". In: *Proceedings of the ACM SIGKDD workshop on outlier detection and description*, pp. 16–21 (cit. on pp. 22, 29, 59).

— (2015). "A meta-analysis of the anomaly detection problem". In: *arXiv preprint arXiv:1503.01158* (cit. on pp. 17, 22, 25, 27–29, 49).

Ferdosi, Bilkis Jamal and Muhammad Masud Tarek (2019). "Visual verification and analysis of outliers using optimal outlier detection result by choosing proper algorithm and parameter". In: *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 2*. Springer, pp. 507–517 (cit. on p. 33).

Feurer, Matthias, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter (2022). “Auto-sklearn 2.0: Hands-free automl via meta-learning”. In: *The Journal of Machine Learning Research* 23.1, pp. 11936–11996 (cit. on p. 32).

Feurer, Matthias, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter (2015). “Efficient and robust automated machine learning”. In: *Advances in neural information processing systems* 28 (cit. on p. 32).

Feurer, Matthias, Jost Springenberg, and Frank Hutter (2015). “Initializing bayesian hyperparameter optimization via meta-learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1 (cit. on pp. 32, 64, 72).

Fisch, Adam, Tommi Jaakkola, and Regina Barzilay (2022). “Calibrated selective classification”. In: *arXiv preprint arXiv:2208.12084* (cit. on p. 38).

Fusi, Nicolo, Rishit Sheth, and Melih Elibol (2018). “Probabilistic matrix factorization for automated machine learning”. In: *Advances in neural information processing systems* 31 (cit. on p. 37).

Goldstein, Markus (2015). *Unsupervised Anomaly Detection Benchmark*. Version DRAFT VERSION. DOI: 10.7910/DVN/OPQMVF (cit. on p. 29).

Goldstein, Markus and Andreas Dengel (2012). “Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm”. In: *KI-2012: poster and demo track* 9 (cit. on pp. 25, 59).

Goldstein, Markus and Seiichi Uchida (Apr. 2016). “A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data”. In: *PLOS ONE* 11.4. Ed. by Dongxiao Zhu, e0152173. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0152173 (cit. on pp. 14, 17, 22, 23, 25, 27, 29, 35, 41, 46, 49, 59).

Grubbs, Frank E. (1969). “Procedures for Detecting Outlying Observations in Samples”. In: *Technometrics* 11.1, pp. 1–21. DOI: 10.1080/00401706.1969.10490657 (cit. on p. 21).

Gudur, Gautham Krishna, R Raaghul, K Adithya, and Shrihari Vasudevan (2022). “Data-Efficient Automatic Model Selection in Unsupervised Anomaly Detection”. In: *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 1443–1448 (cit. on pp. 33, 34).

Guo, Bingjun, Lei Song, Taisheng Zheng, Haoran Liang, and Hongfei Wang (2019). "A Comparative Evaluation of SOM-based Anomaly Detection Methods for Multivariate Data". In: *2019 Prognostics and System Health Management Conference (PHM-Qingdao)*. IEEE, pp. 1–6 (cit. on p. 46).

Gutierrez-Rodríguez, Andres E, Santiago E Conant-Pablos, José C Ortiz-Bayliss, and Hugo Terashima-Marín (2019). "Selecting meta-heuristics for solving vehicle routing problems with time windows via meta-learning". In: *Expert Systems with Applications* 118, pp. 470–481 (cit. on p. 32).

Gutowska, Małgorzata, Suzanne Little, and Andrew McCarren (2023). "Constructing a meta-learner for unsupervised anomaly detection". In: *IEEE Access* (cit. on p. 92).

Guyon, Isabelle, Lisheng Sun-Hosoya, Marc Boullé, Hugo Jair Escalante, Sergio Escalera, Zhengying Liu, Damir Jajetic, Bisakha Ray, Mehreen Saeed, Michèle Sebag, et al. (2019). "Analysis of the automl challenge series". In: *Automated Machine Learning*, p. 177 (cit. on p. 32).

Haibo, He and Ma Yunqian (2013). "Imbalanced learning: foundations, algorithms, and applications". In: *Wiley-IEEE Press* 1, p. 27 (cit. on p. 73).

Han, Changhee, Leonardo Rundo, Kohei Murao, Tomoyuki Noguchi, Yuki Shimahara, Zoltán Ádám Milacski, Saori Koshino, Evis Sala, Hideki Nakayama, and Shin'ichi Satoh (2021). "MADGAN: unsupervised medical anomaly detection GAN using multiple adjacent brain MRI slice reconstruction". In: *BMC bioinformatics* 22.2, pp. 1–20 (cit. on p. 26).

Hawkins, Douglas M. (1980). *Identification of Outliers*. Springer Netherlands. DOI: 10.1007/978-94-015-3994-4 (cit. on p. 21).

He, Zengyou, Xiaofei Xu, and Shengchun Deng (2003). "Discovering cluster-based local outliers". In: *Pattern recognition letters* 24.9-10, pp. 1641–1650 (cit. on p. 27).

Horváth, Tomáš, Rafael G Mantovani, and André CPLF de Carvalho (2016). "Effects of random sampling on svm hyper-parameter tuning". In: *International Conference on Intelligent Systems Design and Applications*. Springer, pp. 268–278 (cit. on pp. 32, 64, 72).

Hutter, Frank, Lars Kotthoff, and Joaquin Vanschoren (2019). *Automated machine learning: methods, systems, challenges*. Springer Nature (cit. on pp. 15, 32).

Jain, Nishant and Pradeep Shenoy (2022). “Selective classification using a robust meta-learning approach”. In: *arXiv preprint arXiv:2212.05987* (cit. on p. 38).

Kanda, Jorge, Andre De Carvalho, Eduardo Hruschka, Carlos Soares, and Pavel Brazdil (2016). “Meta-learning to select the best meta-heuristic for the traveling salesman problem: A comparison of meta-features”. In: *Neurocomputing* 205, pp. 393–406 (cit. on pp. 32, 72).

Kandanaarachchi, Sevvandi, Mario A. Muñoz, Rob J. Hyndman, and Kate Smith-Miles (Mar. 2020). “On normalization and algorithm selection for unsupervised outlier detection”. In: *Data Mining and Knowledge Discovery* 34.2, pp. 309–354. ISSN: 1573756X. DOI: 10.1007/s10618-019-00661-z (cit. on pp. 16, 17, 29, 34, 36, 37, 40–43, 49, 51, 59).

Kaplan, M. Oguz and S. Emre Alptekin (2020). “An improved bigan based approach for anomaly detection”. In: *Procedia Computer Science* 176, pp. 185–194 (cit. on p. 26).

Khan, Irfan, Xianchao Zhang, Mobashar Rehman, and Rahman Ali (2020). “A literature survey and empirical study of meta-learning for classifier selection”. In: *IEEE Access* 8, pp. 10262–10281 (cit. on pp. 15, 32, 33, 35).

Khan, Shehroz S. and Amir Ahmad (2018). “Relationship between variants of one-class nearest neighbors and creating their accurate ensembles”. In: *IEEE Transactions on Knowledge and Data Engineering* 30.9, pp. 1796–1809 (cit. on p. 46).

Kingma, Diederik P and Jimmy Ba (2014). “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (cit. on pp. 61, 68).

Kingma, Diederik P and Max Welling (2013). “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (cit. on p. 60).

Komer, Brent, James Bergstra, and Chris Eliasmith (2014). “Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn”. In: *ICML workshop on AutoML*. Vol. 9. Citeseer, p. 50 (cit. on pp. 32, 64, 72).

Kotlar, Miloš, Marija Punt, Zaharije Radivojević, Miloš Cvetanović, and Veljko Milutinović (2021). "Novel meta-features for automated machine learning model selection in anomaly detection". In: *IEEE Access* 9, pp. 89675–89687 (cit. on pp. 16, 36, 72).

Kotthoff, Lars, Chris Thornton, Holger H Hoos, Frank Hutter, and Kevin Leyton-Brown (2019). "Auto-WEKA: Automatic model selection and hyperparameter optimization in WEKA". In: *Automated machine learning*. Springer, Cham, pp. 81–95 (cit. on p. 32).

Kriegel, Hans-Peter, Matthias Schubert, and Arthur Zimek (2008). "Angle-based outlier detection in high-dimensional data". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 444–452 (cit. on pp. 27, 59).

Le Clei, Constantin, Yasha Pushak, Fatjon Zogaj, Moein Owhadi Kareshk, Zahra Zohrevand, Robert Harlow, Hesam Fathi Moghadam, Sungpack Hong, and Hassan Chafi (2022). "N-1 Experts: Unsupervised Anomaly Detection Model Selection". In: *First Conference on Automated Machine Learning (Late-Breaking Workshop)* (cit. on pp. 17, 34).

Lemke, Christiane, Marcin Budka, and Bogdan Gabrys (2015). "Metalearning: a survey of trends and technologies". In: *Artificial intelligence review* 44.1, pp. 117–130 (cit. on pp. 15, 32).

Li, Yuening, Daochen Zha, Praveen Venugopal, Na Zou, and Xia Hu (2020). "Pyodds: An end-to-end outlier detection system with automated machine learning". In: *Companion Proceedings of the Web Conference 2020*, pp. 153–157 (cit. on p. 33).

Li, Zheng, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu (2020). "COPOD: copula-based outlier detection". In: *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, pp. 1118–1123 (cit. on pp. 25, 59).

Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou (2008). "Isolation forest". In: *2008 eighth ieee international conference on data mining*. IEEE, pp. 413–422 (cit. on pp. 25, 27, 51, 59).

Liu, Yezheng, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He (2019). "Generative adversarial active learning for unsupervised outlier detection". In: *IEEE Transactions on Knowledge and Data Engineering* 32.8, pp. 1517–1528 (cit. on pp. 26, 60).

Mantovani, Rafael G, André LD Rossi, Joaquin Vanschoren, Bernd Bischl, and André CPLF Carvalho (2015). "To tune or not to tune: recommending when to adjust SVM hyper-parameters via meta-learning". In: *2015 International Joint Conference on Neural Networks (IJCNN)*. Ieee, pp. 1–8 (cit. on p. 32).

Mantovani, Rafael G., André L.D. Rossi, Edesio Alcobaça, Joaquin Vanschoren, and André C.P.L.F. de Carvalho (2019). "A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves SVM classifiers". In: *Information Sciences* 501, pp. 193–221. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2019.06.005 (cit. on p. 32).

Mantovani, Rafael Gomes, André Luis Debiaso Rossi, Edesio Alcobaça, Jadson Castro Gertrudes, Sylvio Barbon Junior, and André Carlos Ponce de Leon Ferreira de Carvalho (2020). "Rethinking Default Values: a Low Cost and Efficient Strategy to Define Hyperparameters". In: *ArXiv* abs/2008.00025 (cit. on p. 32).

McCulloch, C.E. and S.R. Searle (2004). *Generalized, Linear, and Mixed Models*. Wiley Series in Probability and Statistics. Applied Probabil. Wiley. ISBN: 9780471654049 (cit. on p. 70).

Meidan, Yair, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici (2018). "N-baiot—network-based detection of iot botnet attacks using deep autoencoders". In: *IEEE Pervasive Computing* 17.3, pp. 12–22 (cit. on pp. 41, 47, 49).

Mendoza, Hector, Aaron Klein, Matthias Feurer, Jost Tobias Springenberg, Matthias Urban, Michael Burkart, Maximilian Dippel, Marius Lindauer, and Frank Hutter (2019). "Towards automatically-tuned deep neural networks". In: *Automated machine learning*. Springer, Cham, pp. 135–149 (cit. on p. 32).

Muñoz, Mario A, Yuan Sun, Michael Kirley, and Saman K Halgamuge (2015). "Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges". In: *Information Sciences* 317, pp. 224–245 (cit. on pp. 15, 32).

Olson, Randal S and Jason H Moore (2016). “TPOT: A tree-based pipeline optimization tool for automating machine learning”. In: *Workshop on automatic machine learning*. PMLR, pp. 66–74 (cit. on p. 32).

Papastefanopoulos, Vasilis, Pantelis Linardatos, and Sotiris Kotsiantis (2021). “Unsupervised Outlier Detection: A Meta-Learning Algorithm Based on Feature Selection”. In: *Electronics* 10.18, p. 2236 (cit. on pp. 17, 34).

Park, Yookoon and David M Blei (2023). “Density Uncertainty Layers for Reliable Uncertainty Estimation”. In: *arXiv preprint arXiv:2306.12497* (cit. on p. 38).

Pevnỳ, Tomáš (2016). “Loda: Lightweight on-line detector of anomalies”. In: *Machine Learning* 102.2, pp. 275–304 (cit. on p. 27).

Pham, Tuan, Rob Hess, Crystal Ju, Eugene Zhang, and Ronald Metoyer (2010). “Visualization of diversity in large multivariate data sets”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6, pp. 1053–1062 (cit. on pp. 44, 45).

Prudêncio, Ricardo BC and Telmo M Silva Filho (2022). “Explaining Learning Performance with Local Performance Regions and Maximally Relevant Meta-Rules”. In: *Brazilian Conference on Intelligent Systems*. Springer, pp. 550–564 (cit. on p. 38).

Ramaswamy, Sridhar, Rajeev Rastogi, and Kyuseok Shim (2000). “Efficient algorithms for mining outliers from large data sets”. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 427–438 (cit. on pp. 24, 27, 59).

Rayana, Shebuti (2016). *ODDS Library* (cit. on p. 29).

Rice, John R (1976). “The algorithm selection problem”. In: *Advances in computers*. Vol. 15. Elsevier, pp. 65–118 (cit. on pp. 15, 32, 55).

Ruff, Lukas, Jacob R. Kauffmann, Robert A. Vandermeulen, Gregoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus Robert Muller (May 2021). “A Unifying Review of Deep and Shallow Anomaly Detection”. In: *Proceedings of the IEEE* 109.5, pp. 756–795. ISSN: 15582256. DOI: 10.1109/JPROC.2021.3052449 (cit. on pp. 14, 21, 23, 28, 48).

Sakurada, Mayu and Takehisa Yairi (2014). "Anomaly detection using autoencoders with nonlinear dimensionality reduction". In: *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, pp. 4–11 (cit. on p. 26).

Sanders, Samantha and Christophe G. Giraud-Carrier (2017). "Informing the Use of Hyperparameter Optimization Through Metalearning". In: *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 1051–1056 (cit. on p. 32).

Schölkopf, Bernhard, Robert C. Williamson, Alex J. Smola, John Shawe-Taylor, and John C. Platt (2000). "Support vector method for novelty detection". In: *Advances in neural information processing systems*, pp. 582–588 (cit. on pp. 25, 59).

Schubert, Erich, Arthur Zimek, and Hans-Peter Kriegel (2014). "Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection". In: *Data mining and knowledge discovery* 28.1, pp. 190–237 (cit. on pp. 23, 56).

Shah, Abhin, Yuheng Bu, Joshua K Lee, Subhro Das, Rameswar Panda, Prasanna Sattigeri, and Gregory W Wornell (2022). "Selective regression under fairness criteria". In: *International Conference on Machine Learning*. PMLR, pp. 19598–19615 (cit. on pp. 38, 76).

Shannon, Claude Elwood (1948). "A mathematical theory of communication". In: *The Bell system technical journal* 27.3, pp. 379–423 (cit. on p. 44).

Shyu, Mei-Ling, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang (Jan. 2003). "A novel anomaly detection scheme based on principal component classifier". In: *Proceedings of the IEEE foundation and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM03)*, pp. 172–179 (cit. on p. 60).

Smith-Miles, Kate A (2009). "Cross-disciplinary perspectives on meta-learning for algorithm selection". In: *ACM Computing Surveys (CSUR)* 41.1, pp. 1–25 (cit. on pp. 15, 16, 32).

Steinruecken, Christian, Emma Smith, David Janz, James Lloyd, and Zoubin Ghahramani (2019). "The automatic statistician". In: *Automated Machine Learning*. Springer, Cham, pp. 161–173 (cit. on p. 32).

Stroup, W.W. (2012). *Generalized Linear Mixed Models: Modern Concepts, Methods and Applications*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis. ISBN: 9781439815120 (cit. on pp. 70, 71).

Tang, Jian, Zhixiang Chen, Ada Wai-Chee Fu, and David W Cheung (2002). "Enhancing effectiveness of outlier detections for low density patterns". In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer, pp. 535–548 (cit. on pp. 27, 59).

Vapnik, Vladimir (2013). *The nature of statistical learning theory*. Springer science & business media (cit. on p. 25).

Vilalta, Ricardo and Youssef Drissi (2002). "A perspective view and survey of meta-learning". In: *Artificial intelligence review* 18.2, pp. 77–95 (cit. on p. 32).

Viola, Rémi, Léo Gautheron, Amaury Habrard, and Marc Sebban (2022). "MetaAP: A meta-tree-based ranking algorithm optimizing the average precision from imbalanced data". In: *Pattern Recognition Letters* 161, pp. 161–167 (cit. on p. 73).

Wang, Chao, Hui Gao, Zhen Liu, and Yan Fu (2018). "A new outlier detection model using random walk on local information graph". In: *IEEE Access* 6, pp. 75531–75544 (cit. on p. 23).

Wang, Hongzhi, Mohamed Jaward Bah, and Mohamed Hammad (2019). "Progress in outlier detection techniques: A survey". In: *Ieee Access* 7, pp. 107964–108000 (cit. on pp. 14, 15, 22, 26–28, 30, 35, 46, 47, 59).

Wiener, Yair and Ran El-Yaniv (2012). "Pointwise tracking the optimal regression function". In: *Advances in Neural Information Processing Systems* 25 (cit. on p. 38).

Wistuba, Martin, Nicolas Schilling, and Lars Schmidt-Thieme (2016). "Two-stage transfer surrogate model for automatic hyperparameter optimization". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*. Springer, pp. 199–214 (cit. on p. 32).

Wolpert, David H and William G Macready (1997). "No free lunch theorems for optimization". In: *IEEE transactions on evolutionary computation* 1.1, pp. 67–82 (cit. on p. 18).

Yang, Xingwei, Longin Jan Latecki, and Dragoljub Pokrajac (2009). "Outlier detection with globally optimal exemplar-based GMM". In: *Proceedings of the 2009 SIAM international conference on data mining*. SIAM, pp. 145–154 (cit. on p. 26).

Yao, Rong, Chongdang Liu, Linxuan Zhang, and Peng Peng (2019). "Unsupervised anomaly detection using variational auto-encoder based feature extraction". In: *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, pp. 1–7 (cit. on p. 46).

Zaoui, Ahmed, Christophe Denis, and Mohamed Hebiri (2020). "Regression with reject option and application to knn". In: *Advances in Neural Information Processing Systems* 33, pp. 20073–20082 (cit. on pp. 38, 39, 76).

Zha, Daochen, Kwei-Herng Lai, Mingyang Wan, and Xia Hu (2020). "Meta-AAD: Active anomaly detection with deep reinforcement learning". In: *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, pp. 771–780 (cit. on p. 34).

Zhang, Ruyi, Yijie Wang, Hongzuo Xu, and Haifang Zhou (2022). "Factorization Machine-based Unsupervised Model Selection Method". In: *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, pp. 796–802 (cit. on pp. 35, 37).

Zhao, Yue, Zain Nasrullah, and Zheng Li (2019). "PyOD: A Python Toolbox for Scalable Outlier Detection". In: *Journal of Machine Learning Research* 20.96, pp. 1–7 (cit. on p. 60).

Zhao, Yue, Ryan Rossi, and Leman Akoglu (2020). *Automating Outlier Detection via Meta-Learning (MetaOD)* (cit. on p. 70).

— (2021). "Automatic unsupervised outlier model selection". In: *Advances in Neural Information Processing Systems* 34, pp. 4489–4502 (cit. on pp. 17, 35, 37, 49, 59, 62, 67, 70, 91).

Zhou, Chong and Randy C. Paffenroth (2017). “Anomaly detection with robust deep autoencoders”. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 665–674 (cit. on p. 26).