Reliable and fair machine learning for risk assessment

Pereira Barata, A.P.

Citation

Pereira Barata, A. P. (2023, April 5). *Reliable and fair machine learning for risk assessment*. *SIKS Dissertation Series*. Retrieved from https://hdl.handle.net/1887/3590289

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/3590289

Note: To cite this publication please use the final published version (if applicable).

RELIABLE AND FAIR MACHINE LEARNING FOR RISK ASSESSMENT

António Pereira Barata

Human Environment and Transport
Inspectorate
Ministry of Infrastructure
and Water Management

Universiteit
Leiden
The Netherlands

The work in this book was funded by the Ministry of Infrastructure and Water Management through Leiden University. The author used facilities at the Leiden Centre of Data Science, the Leiden Institute of Advanced Computer Science, and the Innovation and Data Lab of the Human Environment and Transport Inspectorate.

SIKS

SIKS Dissertation Series No. 2023-06
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Copyright ©2023 António Pereira Barata
Without written permission of the author, no part of this thesis may be reproduced, stored, or published in any form.

Typeset using LaTeX, figures generated using `Matplotlib`
Printed by: Gildeprint, Enschede, the Netherlands, gildeprint.nl
ISBN: 978-94-6419-719-8

Reliable and Fair Machine Learning for Risk Assessment

Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden,
op gezag van rector magnificus prof.dr.ir. H. Bijl,
volgens besluit van het college voor promoties
te verdedigen op woensdag 5 april 2023
klokke 13:45 uur

door

António Pereira Barata
geboren te Lisboa, Portugal
in 1989

Promotor:	Prof.dr. H. J. van den Herik	
Co-promotores:	Dr. C.J. Veenman	
	Dr. F.W. Takes	
Promotiecommissie:	Prof.dr. T.H.W. Bäck	
	Prof.dr. C.M. Jonker	(TU Delft)
	Prof.dr. M.V. Dignum	(Umeå University)
	Prof.dr. J.N. Kok	(Universiteit Twente)
	Prof.dr. H.H. Hoos	(RWTH Aachen University)

Faculteit Wiskunde en Natuurwetenschappen

Dedicated to Alexa

"It is not your fault, but it is your responsibility."

Cor J. Veenman

Contents

List of Abbreviations **xiii**

List of Definitions **xv**

List of Figures **xvii**

List of Tables **xix**

1 Introduction **1**
1.1 The Inspectorate 3
1.2 Machine Learning Preliminaries 5
1.3 Data Quality 7
1.3.1 Missingness 8
1.3.2 Noise 10
1.4 Fairness 12
1.4.1 Performance-Fairness Trade-Off 14
1.4.2 Addressing Fairness in Machine Learning 15
1.5 Problem Statement and Research Questions 16
1.5.1 Problem Statement 16
1.5.2 Research Questions 16
1.5.3 Research Methodology 17
1.6 Research Contributions 18
1.7 Thesis Overview 20

2 Imputation versus Missing-Indicator 23
2.1 Minimising the Impact of Missing Data 24
2.2 Missing Data 26
    2.2.1 Mechanisms of Missingness 26
    2.2.2 Synthesising Missing Values 28
2.3 Related Work 28
2.4 Method 30
    2.4.1 Missing-Indicator 30
    2.4.2 Data 30
    2.4.3 Value Removal 32
    2.4.4 Imputers and Classifiers 32
2.5 Experiments 34
    2.5.1 Experimental Setup 34
    2.5.2 Results 35
    2.5.3 Discussion 38
2.6 Chapter Conclusion 39

3 Crosslier Detection 43
3.1 Crossliers and Miscategorisation 44
3.2 Problem Description 46
3.3 Related Work 47
    3.3.1 Supervised and Semi-supervised Learning 47
    3.3.2 Unsupervised Learning 48
    3.3.3 Data Quality Assurance 49
3.4 The EXPOSE Method 49
    3.4.1 Classification Setup 50
    3.4.2 Model Selection 50
    3.4.3 Crosslier Score 51
    3.4.4 Crosslier Diagram 52
3.5 Experiments 53
    3.5.1 Waste Transportation Setup 53
    3.5.2 Benchmark Setup 54
3.6 Results 56
    3.6.1 Waste Transportation 56
    3.6.2 Benchmark 58
3.7 Discussion 60
3.8 Chapter Conclusion 60

4 Noise-Resilient Classifier 63
- 4.1 Noise and Performance Degradation . . . 64
- 4.2 Problem Description . . . 67
  - 4.2.1 Noise Interpretation . . . 67
  - 4.2.2 Formal Problem Description . . . 67
- 4.3 Related Work . . . 68
  - 4.3.1 Classifier Learning . . . 68
  - 4.3.2 Label Noise Detection . . . 70
- 4.4 The DENOISE Method . . . 71
  - 4.4.1 Learning with Sample Weights . . . 71
  - 4.4.2 Posterior Estimation and Detection . . . 72
- 4.5 Experiments . . . 74
  - 4.5.1 Data . . . 75
  - 4.5.2 Synthetic Noise . . . 75
  - 4.5.3 Evaluation . . . 76
- 4.6 Results . . . 77
  - 4.6.1 Classifier Learning Task . . . 79
  - 4.6.2 Label Noise Detection Task . . . 81
- 4.7 Chapter Conclusion . . . 82

5 Fair Tree Classifier 85
- 5.1 Algorithmic Fairness . . . 86
- 5.2 Problem Description . . . 87
- 5.3 Related Work . . . 88
  - 5.3.1 Measures of Fairness . . . 88
  - 5.3.2 Fair Tree Splitting Criteria . . . 90
- 5.4 The SCAFF Method . . . 91
  - 5.4.1 AUC Computation . . . 91
  - 5.4.2 Strong Demographic Parity . . . 92
  - 5.4.3 Splitting Criterion AUC For Fairness . . . 93
  - 5.4.4 Tree Construction . . . 95
- 5.5 Experiments . . . 96
  - 5.5.1 Datasets . . . 96
  - 5.5.2 Experimental Setup . . . 97
- 5.6 Results . . . 98
  - 5.6.1 Binary Sensitive Attribute . . . 98
  - 5.6.2 Multiple and Multicategorical Cases . . . 100
  - 5.6.3 Relationship with Demographic Parity . . . 102
- 5.7 Chapter Conclusion . . . 104

6 Conclusions **107**
6.1 Answers to the Research Questions 107
6.2 Answer to the Problem Statement 110
6.3 Future Research 111

References **115**

Summary **135**

Samenvatting **141**

Curriculum Vitae **147**

Publications **149**

Acknowledgements **151**

SIKS Dissertation Series **155**

List of Abbreviations

AI	artificial intelligence
AP	average precision
AUC	area under the receiver operating characteristic curve
CV	cross-validation
EU	European Union
FAHT	fairness-aware Hoeffding tree
FPR	false positive rate
IF	isolation forest
ILT	Inspectie Leefomgeving en Transport
KNNC	k-nearest neighbours classifier
LOF	local outlier factor
LR	logistic regression
MAR	missing at random
MCAR	missing completely at random
MNAR	missing not at random
NAR	noise at random
NCAR	noise completely at random
NNAR	noise not at random
NWF	non-white female
NWM	non-white male
OvR	one-versus-rest
PS	problem statement
RQ	research question
SCAFF	splitting criterion AUC for fairness
SVMC	support vector machine classifier
TPR	true positive rate
WF	white female
WM	white male
XGBC	extreme gradient boosting classifier
XGBR	extreme gradient boosting regression

List of Definitions

1.1 AI systems 1
1.2 Trustworthy AI 2
1.3 Risk assessment 3
1.4 Machine learning 5
1.5 Target function (model) 5
1.6 AUC 7
1.7 Fairness 13
1.8 Demographic parity 13
1.9 Strong demographic parity 13
1.10 Data quality 8
1.11 Missingness 8
1.12 MCAR 9
1.13 MAR 9
1.14 MNAR 9
1.15 Imputation 9
1.16 Missing-indicator 10
1.17 Noise 10
1.18 Feature noise 11
1.19 Label noise 11
1.20 NCAR 11
1.21 NAR 11
1.22 NNAR 12
1.23 CRISP-DM 17

2.1 Class balance 31
2.2 Warm start 33

3.1 Crosslier 43
3.2 Crosslier diagram 43
3.3 Outlier 45
3.4 Anomaly 45
3.5 Crosslyingness 46

4.1 Belongingness 66
4.2 Noisy sample 67
4.3 Sample weighting 69

5.1 Threshold-dependent fairness measure 86
5.2 Threshold-independent fairness measure 87
5.3 Equal opportunity 89
5.4 Equalised odds 89
5.5 Sensitive AUC 93
5.6 Orthogonality parameter Θ 93

List of Figures

1.1 Supervised classifier learning 6
1.2 CRISP-DM 18

2.1 Imputation through missing-indicator 30
2.2 KNNC mean performances across missing proportions 36
2.3 SVMC mean performances across missing proportions 37
2.4 XGBC mean performances across missing proportions 38

3.1 Anomaly, outlier, and crosslier 45
3.2 Crosslier detection 46
3.3 Distinction between outlier and crosslier 48
3.4 Crosslier diagrams of four waste categories 58
3.5 Crosslier detection performance across different methods 59

4.1 Learning performances with noisy data (aggregated) 78
4.2 Detection performances of noisy samples (aggregated) 80

5.1 Effect of the orthogonality parameter Θ over SCAFF scores 94
5.2 Computing AUC values for SCAFF 96
5.3 Performance-fairness across methods per dataset 99
5.4 Performance-fairness for the multiple and intersectional cases . . 101
5.5 Effect of orthogonality Θ over demographic parity. 103

List of Tables

2.1 Three different missing mechanisms 27
2.2 Summary of dataset characteristics 31
2.3 KNNC Wilcoxon p-values 36
2.4 SVMC Wilcoxon p-values 37
2.5 XGBC Wilcoxon p-values 38

3.1 Datasets retrieved for crosslier simulations. 55
3.2 Model performances per waste category. 57

4.1 Datasets retrieved for noise simulations 75
4.2 Learning performances with noisy data (per dataset) 79
4.3 Detection performances of noisy samples (per dataset) 81

5.1 Correlation between (strong) demographic parities 104

Chapter 1

Introduction

Modern society is increasingly reliant on information and communication technologies. This includes machine learning methods and their employment in artificial intelligence (AI) systems which are rapidly becoming indispensable components of the status quo. As these technologies evolve, their societal integration shapes the manner in which aspects of education, health, economy, and government are conducted [Ahirwar, 2020].

Given their broad range in application and inherent automated nature, the implementation of these technologies comes with associated risks; e.g., on democracy, the rule of law, and distributive justice, or on the human mind itself in the form on opinion manipulation. To prevent and minimise such risks, there is currently a focus on the foundations, realisation, and assessment of trustworthy AI in the European Union (EU), under which the definition of AI systems is as follows [European Commission, 2019a].

Definition 1.1 – ***AI systems***

AI systems are software (and possibly also hardware) systems designed by humans that, given a complex goal, act in the physical or digital dimension by perceiving their environment through data acquisition, interpreting the collected structured or unstructured data, reasoning on the knowledge, or processing the information, derived from data and deciding the best action(s) to take to achieve the given goal.

Some AI systems can adapt their behaviour after analysing how the environment is affected by their previous actions. The task of achieving *trustworthy AI* and applying it to our society has been set in motion, as official framework guidelines are nowadays arising. In accordance with the High-Level Expert Group on AI [European Commission, 2019b], the definition of trustworthy AI follows.

Definition 1.2 – ***Trustworthy AI***

Trustworthy AI is, on a foundational level, an AI system which abides to the four ethical *principles* of trustworthiness: (1) respect for human autonomy; (2) prevention of harm; (3) fairness; and (4) explainability.

Yet, these principles are meant as broad ideological statements, rather than objective instructions. As such, towards their realisation, technical and non-technical *methods* must be employed. On the one hand, *technical* methods relate to concepts such as model development and model testing. On the other hand, *non-technical* methods entail codes of conduct at an organisational level of entities [CLAIRE, 2021].

Depending on the related risks, AI systems have more or less stringent obligations which must be followed. Specifically, AI systems of which the deployment may put the life and health of citizens at risk are termed *high-risk* [European Commission, 2021]. Towards their utilisation, high-risk AI systems will be subject to stringent obligations, such as the minimisation of discriminatory outcomes, adequate assessment of the performance of the system, whilst having appropriate human oversight.

The present thesis focuses on the *technical methods* towards trustworthy AI in Europe, specifically for high-risk AI systems in light of the risk assessment activities enacted by the Human Environment and Transport Inspectorate of the Netherlands: *Inspectie Leefomgeving en Transport* (hereinafter Inspectorate or ILT). Below, in Section 1.1, we provide a brief introduction to the Inspectorate and the risk assessment activities therein acted, with focus on the issues associated with the shift towards a data-driven paradigm. Concretely, we will narrow down these issues from a machine learning perspective and address them with respect to: *reliability*, in the form of the quality of data; and *fairness* in the form of bias in data. We will do so prior to defining the problem statement and research questions, as to provide the necessary context to the reader.

Section 1.2 provides the preliminaries of machine learning. In Section 1.3, we describe the problems related to the quality of real-world data, viz. *missingness* and *noise*. Section 1.4 discusses the concerns of learning from biased data (i.e., fairness) in machine learning. The 3 aforementioned Sections present both formal definitions and examples of practical applications in the Inspectorate of those definitions, as a way to make explicit the points which are relevant towards formulating, in Section 1.5, the problem statement (PS) and the research questions (RQs). It is remarked that each RQ has its own research methodology which is explained when the RQ is addressed. In Section 1.6, we list our research goals. Lastly, the outline of the remainder of this thesis is given in Section 1.7.

1.1 The Inspectorate

In the Netherlands, the ILT is the legal supervising entity responsible for improving safety, confidence, and sustainability in regard to transport, infrastructure, environment and housing. Practical limitations make it impossible to check the compliance of every single aspect of these broad domains.

Consider, for example, the inspection of ships in the port of Rotterdam, arguably the largest and busiest port in Europe. Every year, over $120,000$ vessels transit the port from around the globe —sea-going vessels— and within the Netherlands —inland vessels— amounting to circa $450,000,000$ metric tons of goods [Port of Rotterdam Authority, 2021]. The ILT must decide how to mobilise their resources, promoting efficacy, efficiency, and feasibility of compliance ensurance.

To ensure compliance, the entities of interest to the ILT (such as companies) are requested to report about their activities to the Inspectorate. This process generates data, often in tabular form. The data are gathered so that domain experts (i.e., inspectors working at the ILT) may analyse them and prioritise their risk assessment activities accordingly.

Given the volume of data gathered by the ILT, it is not possible for the inspectors to consider these data adequately with their current tool set, which is largely comprised of labour-intensive manual analysis of tabular data. If data are not adequately considered —as is the case now— then the proficiency of the risk assessment and inspection activities have high potential for improvement. The opportunity for improving upon these activities in a *data-driven* manner by utilising *machine learning* methods provides the motivation for this thesis.

Risk Assessment

According to [Rausand, 2013], risk assessment is defined as follows.

Definition 1.3 – ***Risk assessment***

Risk assessment is the joint effort of: (1) recognising and analysing possible future occurrences that could harm people, property, or the environment (i.e., hazard analysis); and (2) judging the acceptability of risk based on analysis and taking influencing factors into account (i.e., risk evaluation).

High risk is often associated with the activities performed by the ILT, such as evaluating infrastructure integrity which may jeopardise the life and health of citizens, as failure to comply may result in dire negative health, safety, and environmental impacts. AI systems which are used in such activities clearly fall under the category of *high-risk* AI [European Commission, 2021].

In risk assessment, adequately selecting a non-compliant entity for inspection is termed *targeting*. Failure to perform targeting is termed *mistargeting* and comes in two forms: (1) a non-compliant entity is not selected; and (2) an entity which is compliant is wrongfully targeted. Although the nature of the noncompliance may be diverse (e.g., ship emissions, waste transportation, and infrastructure integrity), mistargeting has dire environmental, health, and safety consequences (type 1 mistargetting) and negatively impacts resources while needlessly disturbing the inspected party (type 2 mistargetting). To mitigate these concerns, minimising mistargeting is paramount (see Section 1.3). To improve on risk assessment in a *data-driven* manner, data are required.

The *quality* of data presents difficult challenges towards implementing data-driven solutions, concretely in the form of *missingness* and *noise* in data. Missingness and noise are known to deteriorate the performance of learned models [Sidi et al., 2012]. On the one hand, high quality data are seldom assured in real world applications, since no data generation method is impervious to flaws (e.g., human entry errors or faulty automated sensors). On the other hand, issues related to low quality data are, in themselves, of particular interest in risk assessment: what might be perceived as low quality data may in actuality be noncompliant behaviour. For example, since different costs are associated with the transport of specific waste materials, companies have financial incentive to purposefully manipulate their transport reports.

In their daily activities, inspectors consider a plethora of factors, together with their domain knowledge, leading to risk assessment decisions. However, not all factors contribute equally to the decision-making process. The prioritisation of ships for inspection via country flag is a case in point.

Traditionally, the country flag of a ship is considered as a proxy for inspection priority: ships sailing under specific country flags are more prone to inspection than other ships with other country flags according to a colour coding —white, grey, or black— based on the detention ratio of ships for that country [Paris MoU, 2020]. The flag is a problem for at least two reasons. First, ships may easily change flags, which allows companies to circumvent risk assessment protocols and elude inspection [Cariou and Wolff, 2011]. Second, the colour of the flag might disproportionally influence the inspection process, which may lead to confirmation bias.

Data represents the *administrative* reality of its encapsulating domain. In other words, they do not necessarily represent the *actual* reality: following from the ship inspection example, most inspectors prioritise high-risk country flags, which will generate data *biased* with respect to that selection. Should a data-driven *tool* (or *model*) be generated from a biased representation of the world, the tool itself may also be biased. Techniques must therefore be employed which reduce bias in models learned from biased data (i.e., learning *fair* models).

1.2 Machine Learning Preliminaries

The term *machine learning* was first introduced in the work by [Samuel, 1959], in which a computer was programmed to learn to play the game of checkers. A general definition would later be proposed by [Mitchell, 1997] as follows.

Definition 1.4 – ***Machine learning***

Machine learning is the study of computer algorithms that improve automatically through experience. A program is said to learn from experience E with respect to task T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Although immense progress has been made since the introduction of the term *machine learning* —from a game of checkers to speech recognition, computer vision, and fraud detection, to name a few— the general definition still holds.

The responsibility of learning a task is delegated to, what is commonly referred to as, a learning algorithm or *learner*: a set of instructions, under which a *loss* —conversely, *gain*— function is either minimised or maximised, respectively. The learning process should culminate in finding the *target function* (i.e., *model*) which translates to the task being solved, herein defined.

Definition 1.5 – ***Target function (model)***

The target function (or model) is the learned function which, provided an input, returns an output which solves the task for which it was learned.

For distinct tasks, specific learners and loss functions may be used. Consider the case in which inspectors must select which ships to inspect from a myriad of vessels. The problem may then be posed as:

Given the characteristics of a vessel, should it be inspected?

In this case, the task is to predict whether or not there is a motive to inspect the ship. The goal is to learn the target function f (or model), of which the *input* x is some vector representation of a vessel and the output y is a *class label* indicating the decision to either $\mathit{inspect}$ $(+)$ or $\mathit{not\text{-}inspect}$ $(-)$. Formally, the target function may be given as $f : x \in X \to y \in Y \subseteq \{+, -\}$, and finding such a function is generally referred to as the *classification* problem [James et al., 2013]. Under a *supervised* learning scenario, learning occurs from a set of observations of which the class labels are known, such that new observations may be classified.

Yet, f may not immediately output a class label $\{+, -\}$. Instead, the output may be given as a *classification score* proportional to the *posterior class probability* $f(x) \propto P(y|x)$. By applying a threshold t to $f(x)$, a class prediction $\hat{y}$ is induced.

The class is predicted as either positive "+" if $f(x) \geq t$, or negative "−" if $f(x) < t$. For simplicity, $\hat{y}_+$ and $\hat{y} = +$ are equivalent; the same holds for (a) $\hat{y}_-$ and $\hat{y} = -$, (b) y_+ and $y = +$, and (c) y_- and $y = -$.

An example of supervised classifier learning as a solution to the classification problem is illustrated in Fig. 1.1. Here we see the combination of a class label and a feature vector. The class label of each sample is represented by colour: red indicating a positive class label y_+, and blue indicating a negative class label y_-. The feature vector of a sample is, in this case, of length 2 and is represented as a point of which the coordinates are the values of each feature: *Feature 1* is the horizontal axis and *Feature 2* is the vertical axis.

A solution to this classification problem example is given in the form of the learned target function $f(x)$, of which the output is represented as a colour gradient in the feature space (i.e., graph) and colour bar: a solid red colour indicates a high classification score, a white colour indicates a classification score of 0, and a solid blue colour indicates a low classification score. Class predictions can then be induced for unseen observations by considering the threshold $t = 0$, marked as a dotted line: if $f(x) \geq 0$, then the class label is predicted as positive ($\hat{y}_+$); otherwise negative ($\hat{y}_-$).

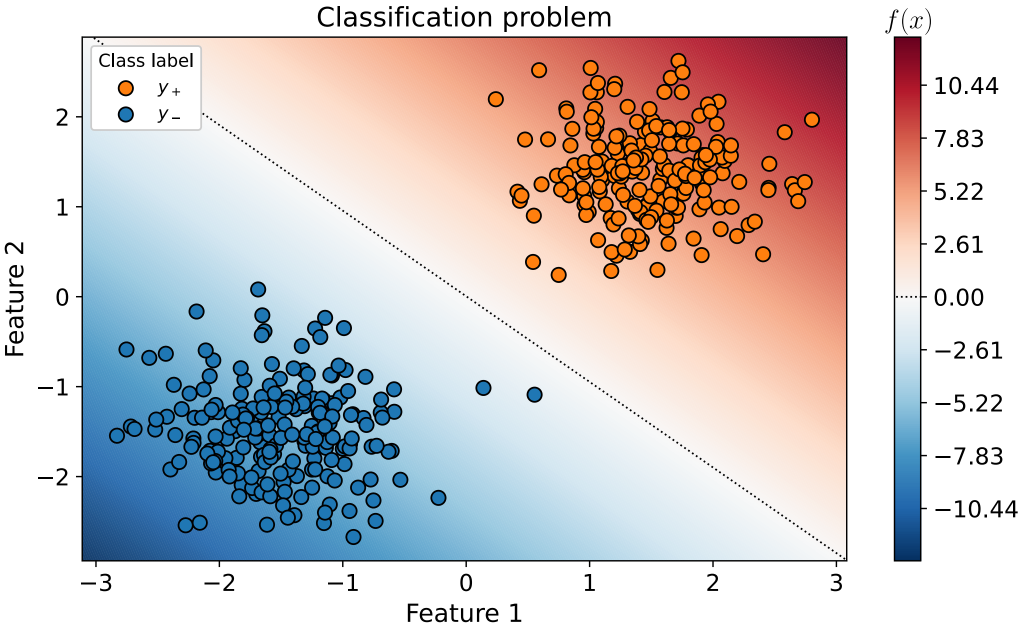


Figure 1.1: **Supervised classifier learning.** The colour gradient represents the classification score $f(x)$ of a classifier learned on the observations shown. The dotted line indicates the threshold $t = 0$ which induces a class label prediction for unseen observations: positive if $f(x) \geq 0$, and negative otherwise.

The performance of a model is estimated on data which was not used for learning, simulating the new (unobserved) real-world observations. These disjoint sets are typically denoted as the *train* (or *learning)* set and the *test* set. Multiple train-test splits (also known as folds) may be used such that the average performance across test sets is computed. This is termed *cross-validation* (CV) and is commonly applied to compute the expected performance of the final deployed model [Stone, 1974].

In the literature, the Area Under the receiver operating characteristic Curve (AUC) [Hanley and McNeil, 1982] is the standard used to measure the performance of a classification model [Flach, 2016]. It is defined as follows.

Definition 1.6 – ***AUC***

AUC is a measure of classification performance which considers the ranking of the output of a model. It quantifies the class separability of a learned model and is given as the probability that, provided a test set, a test sample y_+ selected at random will have a greater classification score than that of a test sample y_- also selected at random.

The curve it plotted in a graph by considering at each threshold t, the true positive rate (TPR) and the false positive rate (FPR): the vertical axis represents the TPR, and the horizontal axis represents the FPR. The greater the AUC — between 0.5 (random ordering) and 1 (perfect ordering)— the greater the performance. To note, other methods exist to compute the AUC which we address in Chapter 5.

It is known that there is no single learning algorithm best suited for all potential classification problems, referred to as the *no free lunch theorem* [Wolpert and Macready, 1997]. Yet, we note that for tabular data —the type of data handled by inspectors— decision tree learning algorithms [Breiman et al., 1984] are known to produce well-performing models —even when learning from low quality data— when applied under bagging (i.e., random forests) or (gradient) boosting strategies [Dogru and Subasi, 2018]. Taking these notions into consideration, the thesis focuses on decision tree learning algorithms.

1.3 Data Quality

The quality of the data used to learn a model often impacts the performance of the *downstream* (or *ulterior*) task of the model (e.g., targeting noncompliance in risk assessment). As per [Fürber, 2016], data quality is defined as follows.

Definition 1.7 – ***Data quality***

Data quality is the degree to which data fulfil requirements. The requirements can thereby be defined by (1) quality requirements of several different individuals of groups of individuals, (2) standards, (3) laws and other regulatory requirements, (4) business policies, or (5) expectations of data processing applications.

Following the definition, the fifth requirement is our main data quality provision. Broadly speaking, data are typically considered of either of *high quality* or *low quality* if they are well-suited or ill-suited, respectively, for the intended downstream task. The latter case being often anecdotally referred to as *garbage in, garbage out* [Rose and Fischer, 2011].

Poor data quality may manifest itself differently. For example, learning from data with insufficient sample size leads to a poor-performing model and, hence, a low performance of the downstream task. Specifically in this work, the focus on data quality is in terms of *missingness* and *noise*, given their prevalence in the domain of the ILT. While missingness is the absence of values in data, noise relates to data values which are inconsistent or erroneous [Sidi et al., 2012]. The names of these types of data quality issues are preceded by an *M* (missingness), or an *N* (noise), see below. Towards building reliable models, these issues must be considered.

1.3.1 Missingness

When dealing with real-world data, the absence of feature values in samples is bound to occur. This occurrence is termed missingness in data and is defined as follows [Beale and Little, 1975].

Definition 1.8 – ***Missingness***

Missingness in data is the occurrence of absence (i.e., missing values) in one or more features of one or more samples.

Missingness is characterised according to the relationship between the missing entries, the observed data, and the values which are missing; these relationships are categorised into three mechanisms: (1) missingness completely at random (MCAR); (2) missingness at random (MAR); and (3) missingness not at random (MNAR). We define each of the three mechanisms below, following [Little and Rubin, 2019].

*Definition 1.9 – **MCAR***

MCAR is the mechanism of missingness which assumes that the probability that a data value is missing is the same for all samples and features.

Under this mechanism, there is *neither* a relationship between the missing values and the remainder of the observed (non-missing) entries *nor* a relationship with the missing value itself. This means the distribution of missingness is independent of the data. An example of this mechanism is a sulphur sensor which runs out of power: some of the data will be missing due to a random event. Nevertheless, MCAR is generally atypical when dealing with real-world data.

*Definition 1.10 – **MAR***

MAR is the mechanism of missingness which assumes that the events that lead to missingness are dependent of the feature values in the observed (i.e., non-missing) data.

The MAR mechanism assumes that missing entries have some form of dependency with respect to the observed entries. For instance, if certain ports do not have sulphur sensors, then vessels travelling through those ports will not generate data regarding those sulphur measures.

*Definition 1.11 – **MNAR***

MNAR is the mechanism of missingness which assumes that the missingness is dependent on actual value that is missing as well as the observed values.

The MNAR mechanism occurs when the absence of entries is *dependent* on both (a) the unseen values and (b) the observed data. For example, when companies which fail to report on their emissions are the most likely to have systematically higher emission levels.

Traditionally, learning algorithms are incapable of handling data with missing values [García-Laencina et al., 2010]. The data must first be artificially made *complete*, i.e., without missing values, via *missing data-handling techniques*. In the missingness literature, two prominent categories of missing data-handling techniques are considered [Enders, 2010]: (1) imputation; and (2) missing-indicator. These are defined below. To note, several imputation methods exist in the literature [Enders, 2010]. We elaborate further on this topic in Chapter 2.

*Definition 1.12 – **Imputation***

Imputation is the process of *filling in* missing values based on the available data.

Definition 1.13 – ***Missing-indicator***

Missing-indicator is the method by which missingness is *encoded*, by generating an additional binary feature representing the presence or absence of values, and by assigning the same value to all missing values of the feature of concern.

It is known that (a) the mechanism of missingness, (b) the choice of missing data-handling technique, and (c) the learning algorithm jointly play a crucial role in the final model performance [Garciarena and Santana, 2017]. For instance, under non-MCAR, the missing-indicator method is a viable solution towards classifier learning [Lipton et al., 2016].

Discerning which missing mechanism is present is a challenging task. While it is impossible to distinguish between MAR and MNAR —as the necessary information for the distinction is itself missing— a test for MCAR vs not-MCAR has been proposed [Little, 1988]. Yet, the outcome of the test is not entirely reliable, as false positives and false negatives may still occur.

Real-world data are seldom MCAR [Van Buuren, 2018]. However, testing for MCAR does not provide a guaranteed result. Even though the assumption of non-MCAR data is generally correct in risk assessment, the MCAR mechanism is still possible and not necessarily detectable. Putting it differently, the issue is to find a solution to missingness which mitigate the detriment to the performance of the downstream task in cases where the assumption of non-MCAR is false. We will focus on this issue in Chapter 2.

1.3.2 Noise

In classification, noise in data is defined as follows [Angluin and Laird, 1988].

Definition 1.14 – ***Noise***

Noise in data is the presence of elements in (a) the feature(s) and/or (b) the class label which obscures their relationship and complicates model learning.

The major consequence of noise is the performance degradation of the final learned model when it is ignored [Wilson and Martinez, 2000]. It negatively impacts the performance of the learned model by obscuring the relationship between features and class label. Noise is denominated as either (a) *feature noise* or (b) *(class) label noise*, depending on the elements affected [Sáez et al., 2014]; these denominations are defined below.

*Definition 1.15 – **Feature noise***

Feature noise is the presence of elements in the feature values of samples which obscures the relationship between the features and the class label.

*Definition 1.16 – **Label noise***

Label noise is the presence of erroneous class labels in samples (i.e., mislabels).

Depending on the type of noise (feature or label), the compromise in performance of the downstream classification task varies. In general, we may notice that feature noise tends to be less detrimental than label noise [Zhu and Wu, 2004]. Here we note that in cases where both feature and label noises are present, the noise is denoted as a special case of label noise [Frénay and Verleysen, 2013]. As such, hereinafter, the terms *noise* and *label noise* are synonymous, and are both denoted as N.

Label noise is described according to the relationship between the mislabels and data characteristics. Three *label noise mechanisms* are used to categorise these relationships [Frénay and Verleysen, 2013]: (1) label noise completely at random (NCAR); (2) label noise at random (NAR); and (3) label noise not at random (NNAR). We define each of the mechanisms below.

*Definition 1.17 – **NCAR***

NCAR is the label noise mechanism which assumes that the probability that a sample is mislabelled is the same for all samples.

NCAR occurs when the proportion of mislabels is the same across classes. It is associated with random errors in the data generation process; e.g., automated sensor errors.

*Definition 1.18 – **NAR***

NAR is the label noise mechanism which assumes that the probability that a sample is mislabelled be dependent on the class label of the sample.

NAR entails different proportions of mislabels across the different classes. In other words, samples of one class are more prone to being mislabels than samples from another class. This might result from ill-calibrated tests to determine the outcome of risk assessment; e.g., targeting protocols which are too stringent or too relaxed.

Definition 1.19 – ***NNAR***

NNAR is the label noise mechanism which assumes that the probability that a sample is mislabelled is dependent on the class label and/or the features values.

In NNAR, mislabels may be associate with specific regions of the feature space, and their proportion may or may not be the same across classes. This mechanism is particularly interesting as it relates to the report-manipulation example from Section 1.1.

Since label noise may correlate to noncompliance within the context of risk assessment, especially in conjunction with feature noise, we focus on the NNAR scenario. Addressing label noise towards model *learning* often involves a prerequisite in the form of a sample *detection* step. Generally, label noise detection approaches leverage supervised learning methods into producing mislabelling *detection scores*. Detection scores are used to identify samples such that higher detection scores indicate higher likelihood of mislabel, and can be generated by exploiting classification scores $f(x)$: the lower the classification score of a sample towards its class label, the higher the detection score for being a noisy label. Given that these scores quantify the amount of label noise in samples [Jeatrakul et al., 2010], they may be used to better learn a classification model trained on label-noisy data [Liu and Tao, 2015].

Under the current risk assessment scenario, it would be advantageous to exploit these scores two-fold. First, when label noise mechanisms may translate to non-compliance, detection scores can directly be used as risk assessment scores. Second, detection scores may be incorporated into model learning such that the performance of the final learned classifier is the least compromised by noise, promoting better-performing risk assessment models. The work in this thesis addresses these two topics in Chapter 3 and Chapter 4, respectively.

1.4 Fairness

Machine learning algorithms model all sorts of relations between features and outcomes in historical training data, including potential societal biases [Richardson, 2022]. Within the Inspectorate, confirmation bias in historical inspection data is a case in point (we expand on it in Chapter 5).

The problem is to learn a mostly *unbiased* model from biased data. We mention *mostly* since, as detailed further, a completely unbiased model has a completely random classification output, rendering it useless. Learning with biased data is a problem traditionally termed *fairness* [Barocas et al., 2017] and is defined as follows.

Definition 1.20 – ***Fairness***

Fairness in machine learning is the study and tentative correction of algorithmic bias which results from learning with biased data.

A model is deemed more or less fair if its output has lesser or greater dependency (i.e., bias), respectively, towards some *sensitive* characteristic, such as nationality, age, or gender; i.e., a model is deemed less fair if it *favours* certain groups or individuals over others. To note, although the term *bias* may have different meanings in other fields (e.g., the bias-variance trade-off [Kohavi et al., 1996, Meertens et al., 2021]), in this context it is used antithetically to the term *fairness*.

Without loss of generality, the sensitive attribute of a sample is denoted $s \in S \subseteq \{-, +\}$ and represents sensitive group information, such as gender; the male and female groups are represented as S_+ and S_- or vice-versa, and samples pertaining to each group are respectively s_+ and s_-. Measures of fairness attempt to quantify the disparity of the model output between the groups conditioned on the sensitive attributes. The model output considered may be either (a) the class label prediction $\hat{y}$ induced by a set threshold t or (b) the classification score $f(x)$ [Venkatasubramanian, 2019]. Below we define two prevalent measures of fairness as described in the literature. They consider each of these outputs and are called: (1) *demographic parity* [Feldman et al., 2015] and (2) *strong demographic parity* [Jiang et al., 2020].

The definition of demographic parity follows.

Definition 1.21 – ***Demographic parity***

Demographic parity is the fairness measure which considers the difference in the proportion of positive outcomes (i.e., positive class label predictions) between two sensitive groups S_+ and S_-.

An example of demographic parity is the difference in the proportion of men who are hired vs the proportion of women who are hired. By extending the definition of demographic parity to account for the classification score (instead of the induced class prediction), the measure of strong demographic parity can be defined as follows.

Definition 1.22 – ***Strong demographic parity***

Strong demographic parity is the fairness measure which quantifies the fairness of a learned model by considering the difference in the ranking of classification scores across sensitive groups S_+ and S_-.

The computational definitions of demographic parity and strong demographic parity are given in Chapter 5.

To note, other fairness measures exist which consider different relations between model output and sensitive information; for example, the TPR and/or FPR conditioned on the sensitive groups (i.e., equal opportunity and equalised odds) [Pessach and Shmueli, 2022]. Yet, in the thesis we focus on the aforementioned (strong) demographic parities, given their link between class prediction and classification score. For both demographic parity measures, values closer to 0 indicate *model fairness*, whereas values closer to 1 indicate *model bias* Moreover, we remark that the strong demographic parity is conceptually similar to the AUC performance measure, but it is conditioned on the sensitive attribute values. We make use of this observation in Chapter 5.

1.4.1 Performance-Fairness Trade-Off

There exists a phenomenon under which the following holds. As model fairness increases, the more likely it is that the predictive performance decreases. This is known as the *performance-fairness trade-off* [Zafar et al., 2017]. It is a result of the decorrelation between the features and sensitive attribute, under the assumption of bias in data [Kleinberg et al., 2016].

Nevertheless, the performance-fairness trade-off is not necessarily balanced: greatly improving model fairness does not require a large decrease in model performance. Depending on the dataset, the corresponding correlation between sensitive attributes, and the target variable, it is possible to ensure adequate model fairness with limited decrease in predictive performance. In other words, the trade-off can be leveraged to find the optimal performance-fairness pair of a specific scenario.

The *tunability* of the performance-fairness trade-off in a model should, therefore, be considered. Towards its adequate implementation, we decompose the requirements of the tunability process into the two following: (1) granularity, and (2) intuitiveness.

First, the granularity of the tunability must be implemented such that an optimal performance-fairness pair may be found. If the granularity is insufficient, then the optimal trade-off between performance and fairness may not be reached. To put it differently, by incorporating fine-tuning into the trade-off, it is assured that the *sweet spot* of the performance-fairness is attainable.

Second, the tunability should be incorporated in an intuitive manner towards model usability. Alongside their domain knowledge, relevant stakeholders (in conjunction with the aid of the machine learning expert) should be able to decide which performance-fairness trade-off point is the optimal solution given a specific problem. By making the tunability process accessible, this task-dependent optimality can be assured. We explore the two aforementioned requirements in Chapter 5.

1.4.2 Addressing Fairness in Machine Learning

Taxonomically, three distinct mechanisms have been proposed to address fairness in machine learning [Pessach and Shmueli, 2020]. Each mechanism addresses fairness at different stages of the model learning process: (1) pre-processing; (2) post-processing; and (3) in-processing.

First, *pre-processing* relates to changes made within the training set prior to learning a model; e.g., by manipulating the training set specifically towards the homogenisation of the distributions across the different sensitive groups, making it more difficult for the final learned classifier to distinguish between the sensitive groups [Feldman et al., 2015]. This mechanism is sub-optimal because sample manipulation neglects the dependency over the classification task; i.e., the bias in data may still be exploited [Goldfarb-Tarrant et al., 2020].

Second, *post-processing* relates to changes made to the output of the final trained model, correcting decisions over sensitive groups [Hajian et al., 2015]; for instance, by having different decision thresholds t for each group [Corbett-Davies et al., 2017]. However, using sensitive information as input to determine a final outcome —e.g., hire if male and not-hire if female, for the same model score— is often not viable and potentially illegal under the General Data Protection Regulation in EU law [Goddard, 2017].

Third, *in-processing* encompasses the development and/or modification of classification algorithms. In this manner, models account for both predictive performance and fairness during learning by exploiting the relation between the features and sensitive attributes [Bechavod and Ligett, 2017].

Across the three bias-addressing mechanisms, in-processing is the most prevalent in the current literature, with overall superior classification performance and fairness, and the possibility to adequately tune the trade-off [Kamishima et al., 2012, Goh et al., 2016, Woodworth et al., 2017].

The applicability of the mechanisms relates to the degree of freedom of the developer. With pre-processing, there is only access to the data and not the model or its output; i.e., it is most useful for third party model development. In post-processing, only the output of a model is accessible; e.g., closed source algorithms. In-processing implies full developmental privileges (data, model, and output), allowing the relevant requirements to be combined.

Towards accomplishing the work presented in this thesis, we were allowed access by the ILT to their data. Moreover, full model development privileges were provided, including model output. Since, as stated, we are in full control of the algorithmic development, and given the prevalence and overall superiority of in-processing in the current literature, model fairness is addressed in an in-processing fashion in this thesis.

1.5 Problem Statement and Research Questions

In this thesis we are motivated by the real-world operations of the inspectors of the ILT towards risk assessment, which benefit from data-driven (i.e., machine learning) methodologies.

1.5.1 Problem Statement

The shift towards a data-driven paradigm in the operations of the risk assessment experts harbours considerable concerns given the high-risk profile of their endeavours. Based on this observation, we formulate the following PS.

> **PS:** *How can machine learning methods advance data-driven risk assessment by the Inspectorate in a reliable and fair manner?*

To address the PS, we will decompose it into three tractable RQs.

1.5.2 Research Questions

Missingness is a data-quality issue that impacts the performance of a downstream task on a model learned from a dataset. This impact must be considered, as to minimise the performance decrease during operational deployment. The performance of the downstream task of a model learned on data with missingness may vary depending on the joint selection of (a) the missing data-handling technique —imputation, missing-indicator, complete-case analysis, to name a few— (b) the choice of learning algorithm, and (c) the underlying missing mechanism.

Albeit real-world data —such as the one generated by the ILT— is seldom MCAR, it is still a possibility and testing for it does not provide a guaranteed result. Having made these observations, the first RQ is formulated as follows.

> **RQ1:** *Given data with missing values, which (a) missing data-handling technique and (b) learning algorithm should be jointly selected such that, regardless of the missing mechanism, the detriment to the downstream task performance is minimal when compared to the non-missing (unavailable) case?*

Data permeated with noise is detrimental to model learning. Moreover, the detection of these noisy samples is a prerequisite for model learning with noisy data. While noise in classification is strictly any disruption of the relationship between features distribution and labels, we tackle label noise as mislabels in the data under specific feature distribution conditions (i.e., with additional feature noise). Our reasoning for this is two-fold: (1) this noise may be indicative of noncompliant behaviour in risk assessment; and (2) label noise is more detrimental to model learning than feature noise alone.

It is advantageous to produce noisy-sample detection scores usable for both (a) noncompliance targeting, and (b) model learning. Accordingly, the second RQ is a *compound* one —decomposable into RQ2(a) and RQ2(b)— and follows.

> **RQ2:** *Given data with label noise, how can noisy-samples be (a) adequately detected, and (b) used to learn a well-performing model?*

Fairness in machine learning is paramount to handle biased data. Not only must the learned model exhibit adequate predictive performance, it must also ensure that the predictions are the least impacted by the data bias.

To measure the impact of biased data in the learned model, different fairness measures exist which have an analogous (i.e., corresponding) performance measure. Moreover, the performance-fairness trade-off is a well-known phenomenon and may be exploited to achieve the most fairness increase at the cost of the least performance decrease. To exploit the performance-fairness trade-off, model learning must allow for its tunability. An in-processing approach to fairness enables this. Merging these remarks, we arrive at our third RQ.

> **RQ3:** *How can we, from biased data, learn a model tunable with respect to the performance-fairness trade-off such that the selection of the trade-off point is made intuitive for the relevant stakeholders?*

1.5.3 Research Methodology

To provide an answer to the PS, the work in this thesis follows the well-established CRoss Industry Standard Process for Data Mining (CRISP-DM) [Martínez-Plumed et al., 2019], defined as follows and depicted in Fig 1.2.

Definition 1.23 – ***CRISP-DM***

CRISP-DM is a process model which decomposes a data science process into six phases: (1) business understanding; (2) data understanding; (3) data preparation; (4) modelling; (5) evaluation; and (6) deployment.

Here, we explicitly denote that, despite the subject of this thesis not being data mining, CRISP-DM still offers a valuable approach which helps guide our research. To begin our work, communication between the domain experts, stakeholders, and us researchers was fulcral; jointly, efforts were had to promote the first two phases to be best of our capacity. In this thesis, however, the focus is not in detailing the communication processes, but rather to describe the technical methods developed and their performance; i.e., *data preparation* (3), *modelling* (4), and *evaluation* (5). We further denote that the deployment phase is outside of the scope of this thesis, as it depends not solely on the adequacy of the technical methods, but also on changes at the organisational level.

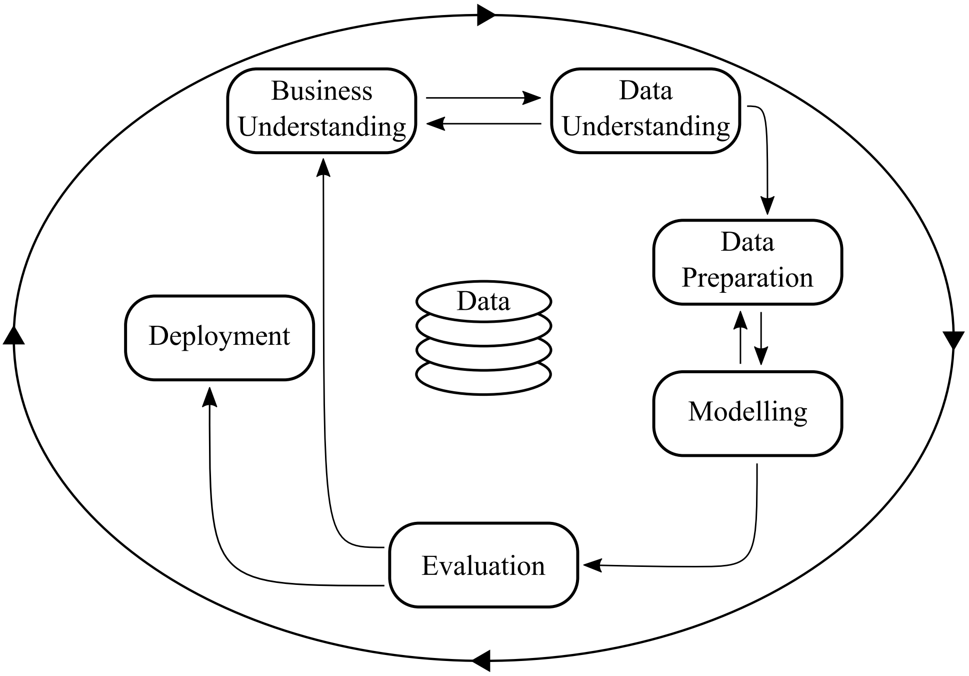


Figure 1.2: **CRISP-DM.** Process model comprised of six sequential phases.

The technical methods used to address RQ1 are detailed in Chapter 2, RQ2(a) in Chapter 3, RQ2(b) in Chapter 4, and RQ3 in Chapter 5.

1.6 Research Contributions

Below, we list the four main contributions of our research.

- *Contribution 1.* We show that towards supervised classifier learning with real-world missing data, a combination of (a) the missing-indicator method and (b) a decision tree learning algorithm —namely, gradient boosting— should be used to minimise the detriment in classification performance.

  According to the literature, non-MCAR scenarios can benefit from the missing-indicator method, measured as the downstream task performance. In the scenario of a non-MCAR assumption being falsely made, we compare several imputation methods to the missing-indicator method, quantifying their differences measured as the downstream classification performance. We empirically demonstrate that across different learning algorithms, (gradient) boosting architectures which incorporate feature selection processes are the least susceptible —if at all— to the sub-optimal decision of applying the missing-indicator method when dealing with data generated under MCAR.

- *Contribution 2.* We propose an approach to targeting and accounting noisy observations which, subsequently, allows for better learning with noisy data, which outperforms the competing literature methods.

  The scenario in which both label and feature noise permeate data is considered. By leveraging an already existing robust decision tree learning algorithm via gradient boosting, noisy-sample detection scores are computed in a CV manner, generating a model with well-calibrated output. Empirically, we performed extensive experimentation to compare our approach to other methods from both outlier and mislabel detection publications. Our novel method —termed EXPOSE— exhibited an overall improved performance over the methods against which it was compared.

- *Contribution 3.* We develop a strategy towards classifier learning for data with label noise through sample weighing which exhibits competitive performance when compared to the current literature, particularly adequate for datasets with large proportions of mislabels.

  Based on the aforementioned contribution in noisy-sample detection, the detection scores are leveraged to compute individual observation weights. The weights are applied within the learning process as coefficients in the logistic loss function. We empirically show that via well-calibrated posterior probability estimations, the *log-odds of an observation* may be leveraged in learning. Through an exhaustive experimental design, comprised of different proportions of both label and feature noise, we validate our proposed method —DENOISE— by comparing it to the state-of-the-art in learning from noisy data under the NNAR scenario. On average, our method achieves superior performance compared to the state-of-the-art.

- *Contribution 4.* We design a fair tree classifier which is independent of threshold in the performance loss as well as the fairness criterion loss. The classifier can be easily adjusted to assess performance-fairness trade-off points.

  The threshold-independent fairness measure of strong demographic parity is used and, by drawing from its analogy to the classification performance measure AUC, we arrive at the Splitting Criterion AUC For Fairness, or SCAFF. Incorporated in SCAFF, the orthogonality parameter Θ which regulates the performance-fairness trade-off. Our learning algorithm considers multiple sensitive attributes simultaneously of which the values may be multicategorical. When compared to other fair tree learning splitting criteria, our experiments with real-world data show our method is able to achieve classification performance and fairness which are on par at worst and superior at best, against those of the competing approaches in the fairness literature.

1.7 Thesis Overview

In Chapter 1, we introduced the current movement towards *trustworthy AI* in Europe. Then, we narrowed our scope towards the particular domain of risk assessment, with distinct concern for data-driven solutions within the Inspectorate of the Netherlands. We further established the required foundations to achieve these solutions. We formulated our PS, and decomposed it into the three RQs of the thesis. Next, we proposed our four contributions. The remainder of the thesis is given below.

- **Chapter 2** answers RQ1, resulting in Contribution 1. The content of the chapter is identical to that of the work by
  Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2019). Imputation methods outperform missing-indicator for data missing completely at random. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 407–414. IEEE

- **Chapter 3** answers RQ2(a). The result of the chapter is Contribution 2. The content of the chapter is identical to that of the work by
  Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2021). The eXPose approach to crosslier detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2312–2319. IEEE

- **Chapter 4** provides an answer to RQ2(b). The result corresponds to Contribution 3. The content of the chapter is identical to that of the work by
  Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2022). Noise-resilient classifier learning. *Pattern Recognition (under review)*

- **Chapter 5** answers RQ3, culminating in Contribution 4. The content of the chapter is identical to that of the work by
  Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2022). Fair tree classifier using strong demographic parity. *Machine Learning (under review)*

- **Chapter 6** entails the conclusions of the present thesis, in three distinct sections. We (1) answer the three research questions, (2) provide an answer to the problem statement, and (3) discuss future work directions.

The papers presented in this thesis were produced as a joint collaboration between the supervisors and the PhD candidate. Discussions on research topics and how to tackle the problems described were addressed as a team. The writing was performed by the candidate, incorporating the commentary provided by the supervisors. The implementation of the experimental designs and gathering of the results were performed by the candidate.

Chapter 2

Imputation versus Missing-Indicator

Missingness is a ubiquitous problem inherent to real-world data. When learning a classifier, missing data is detrimental to the classification performance of the final model. Approaches to deal with missingness can be partitioned into methods that either (a) impute or (b) encode missingness.

Depending on the missing mechanism, some missing data-handling techniques are best suited than others in combination with different learners. Under a non-MCAR mechanism —typical of real-world data— a straightforward approach is to apply the missing-indicator method. However, a non-MCAR missing mechanism is not always guaranteed and testing for it does not ensure a reliable result. In this chapter, we experimentally demonstrate that —under MCAR— the negative impact in downstream classification performance derived from the inadequate application of the missing-indicator can be made identical to that of the application of imputation, particularly by deploying a decision tree-based learning algorithm via gradient boosting.

Therefore, a solution to the problem of missing data is to deploy the missing-indicator method in conjunction with a decision tree-based learner, particularly via gradient boosting, therewith addressing RQ1: given data with missing values, which (a) missing data-handling technique and (b) learning algorithm should be jointly selected such that, regardless of the missing mechanism, the detriment to the downstream task performance is minimal when compared to the non-missing (unavailable) case?

The current chapter corresponds to the following publication:
Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2019). Imputation methods outperform missing-indicator for data missing completely at random. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 407–414. IEEE

2.1 Minimising the Impact of Missing Data

Big data analytics encompasses a multitude of challenges in relation to different data aspects or characteristics. Other than volume, variety, and velocity, the concept of veracity (i.e., data quality) plays a key role when addressing real-world problems. As discussed in [Olson, 2003], the quality of data is intrinsically related to the intended use of the data itself. Moreover, to satisfy this notion of usability, data must be trusted and timely, as well as both accurate and complete. In this work we shall be focusing on the latter mentioned aspect of data (completeness), or rather its conceptual counterpart: data missingness.

The phenomenon of missing data is defined as the absence of observational values within a dataset. It is a widespread obstacle which presents itself in many fields of research where data are analysed, such as econometrics [Dardanoni et al., 2011], psychology [Schlomer et al., 2010], and epidemiology [Pedersen et al., 2017]. Regardless of the underlying reasons for the occurrence of missingness throughout different domains, missing data presents a challenge towards the completion of any data-related task.

The task to be performed after imputation of the dataset is referred to as the downstream task (e.g., regression or classification). Choosing how to handle this issue will influence the outcome of the downstream task. In other words, poor application of missing data-handling techniques leads to underwhelming performance and biased results [Choi et al., 2019]. Thus, depending on the problem to be addressed, it is important to carefully select the most appropriate strategy to overcome missingness and minimise the impact of incomplete data on the final outcome of the downstream task [Little et al., 2014]. Also, the type of data that is missing influences the selection of imputation approaches [Feng et al., 2011].

We know from [Garciarena and Santana, 2017] that the effectiveness of an imputation method in classification is tightly associated to the family of classifiers to be used and the missing mechanism affecting the data. In this context, by *family* we mean a set of classification algorithms of which the decision functions are conceptually similar; i.e., algorithms of which the mappings of the input space into a specific category are alike. For example, a tree-based algorithm recursively splits the original input space into segments through a set of relation operator-based rules, whereas a k-nearest neighbours approach checks the mode of the closest k objects according to some distance metric (often euclidean): we consider the two methods to pertain to different families. By missing mechanism, we are referring to the distribution of missing values in the data. It is common practice to categorise these mechanisms as MCAR, MAR, and MNAR. In the real world, it is only possible to distinguish between MCAR and not-MCAR mechanisms.

Generally, imputation methods rely on statistical concepts (e.g., mean and median) or machine learning approaches (i.e., predictions over missing values). Another commonly used approach to data imputation is the missing-indicator method [Huberman and Langholz, 1999], where a new placeholder value or attribute for missingness is generated to indicate the missing value. Although past studies have been conducted to illustrate how these methods affect the bias of results [Knol et al., 2010], there exist several gaps in the literature, of which we mention three.

First, often the missingness characteristics are not fully reported [Malla et al., 2018]. Second, results can be related to one specific domain rather than to a more general level [García-Laencina et al., 2015]: this leads to conclusions that are either ambiguous or not generalisable with respect to the distinct field task of the reported work. Third, authors tend to disagree on which metrics are best at quantifying imputation effects [Van Buuren, 2018].

Stating that one method of imputation outperforms another method is dependant on the type of performance analysis conducted and the missingness assumptions of mechanisms at play [Santos et al., 2019b]. In general, the purpose of imputation is not making a dataset complete, but rather make it possible to handle data for a specific task. Regardless, research in imputation usually reports performance as a function of error between the artificially removed values and the predicted imputation [Amiri and Jensen, 2016]. This is not a viable metric to compare different imputation methods and the missing-indicator method.

A more realistic approach to measuring the impact of imputation methods is to assess the performance on the post-imputation (downstream) task. Little research has been done in measuring how of the missing-indicator approach to handling missing data performs in classification problems compared to imputation methods [Ding and Simonoff, 2010].

In this work, we compare several imputation methods and the missing-indicator method, and measure their differences on the most relevant measure: classification performance. We establish whether the missing-indicator should or should not be used given the specific case of classification problems with numerical data, under the MCAR mechanism of missingness.

The structure of this chapter is as follows: Section 2.2 will deal with the basic concepts which distinguish different types of missingness and how to synthetically generate missing data. In Section 2.3, we briefly describe the added value of the presented research with respect to past work. Section 2.4 consists of the description of the materials and methods used, while in Section 2.5 we refer to our experimental setup and results. Finally in Section 2.6 our conclusions are given and future research directions are suggested.

2.2 Missing Data

Here we introduce the overall concept of missingness. We illustrate and define each missing mechanism in Section 2.2.1. In Section 2.2.2, we further present how to generate missing values under the specific MCAR assumption.

2.2.1 Mechanisms of Missingness

In research on imputation, missing mechanisms are commonly defined according to the distribution of missing values [Little and Rubin, 2019]. In this manner, data can be missing under three different assumptions: MCAR, MAR, and MNAR. Under MCAR, the probability that a data value is missing is the same for all data points. MAR occurs when the events that lead to missingness are completely at random but only within a subset of some other observed variable within that dataset. Lastly, when neither of the previous two missing mechanisms are at play, but rather the missingness is directly related to the actual value that is missing and/or some other variable value, then the missing mechanism is referred to as MNAR. Table 2.1 illustrates the aforementioned mechanisms. In this example, each row represents an instance in a dataset.

The first column (*Class*) represents some class label, and the remaining columns represent the same feature under different missing mechanisms: *Complete* signifies the observations without any missing values, whereas the three remaining columns illustrate how the different missing mechanisms would affect the set of observations. Each number denotes an observed value.

Following the notation of previous literature [Little and Rubin, 2019], we formally define the different missing mechanisms as follows. Let X be an n by p matrix serving as some dataset with $i = 1, ..., n$ instances and $j = 1, ..., p$ features where $x_{i,j}$ is an individual element of X. Each element $x_{i,j}$ may represent either an observation or a missing value, depending on the characteristics of thee dataset. We can divide X into two disjoint objects, $X = (X^{obs}, X^{miss})$, where X^{obs} and X^{miss} represent the observed and missing values of X. Let M be a matrix of the same shape as X with $m_{i,j} \in M$ where $m_{i,j} = 0$ and $m_{i,j} = 1$ indicate the presence or absence of observation $x_{i,j} \in X$, respectively. Then, the missing mechanism is MCAR if:

$$Pr(M = 1 | X^{obs}, X^{miss}) = Pr(M = 1); \tag{2.1}$$

MAR if:

$$Pr(M = 1 | X^{obs}, X^{miss}) = Pr(M = 1 | X^{obs}); \tag{2.2}$$

and MNAR if:

$$Pr(M = 1 | X^{\text{obs}}, X^{\text{miss}}) = Pr(M = 1 | X^{\text{obs}}, X^{\text{miss}}). \tag{2.3}$$

Table 2.1: Three different missing mechanisms

Class	**Feature**			
	Complete	MCAR	MAR	MNAR
0	18.91	–	–	–
0	13.42	13.42	–	–
0	4.05	–	4.05	4.05
0	4.06	4.06	4.06	4.06
0	18.24	18.24	–	–
0	3.01	–	–	3.01
0	11.37	11.37	11.37	–
0	14.25	–	14.25	–
0	2.74	2.74	2.74	2.74
0	5.24	5.24	–	5.24
0	10.21	–	–	–
1	11.02	11.02	11.02	–
1	7.06	–	7.06	7.06
1	14.29	14.29	14.29	–
1	2.16	–	2.16	2.16
1	5.26	5.26	5.26	5.26
1	0.37	–	0.37	0.37
1	8.24	8.24	8.24	8.24
1	10.36	–	10.36	–
1	6.43	6.43	6.43	6.43
1	1.31	–	1.31	1.31

While it is possible to define these concepts, accurately determining which of these assumptions permeates a dataset is no easy task: the information required to discriminate between MAR and MNAR is, rather unsurprisingly, missing itself. However, such is not the case for MCAR that can be tested for statistically [Little, 1988] albeit false positives and false negatives may still occur.

We know from [Van Buuren, 2018] that specific imputation methods that perform well under some condition might not be applicable under another condition. As such, it is not only important to determine what mechanism shapes the missingness within the data that will be used to address a particular problem, but it is also imperative to report it.

2.2.2 Synthesising Missing Values

Imputation studies rely heavily on generating synthetic missingness [Bertsimas et al., 2017]. The removal of observations can be labelled as either univariate or multivariate, depending on the number of features selected to have their observations deleted by some percentage. The synthesis of missingness varies depending on the target missing mechanism to be implemented, as different conditions have to be met to satisfy specific occurrences of missing values.

Under MCAR univariate missingness generation, selecting the feature of which values will be removed can be performed either randomly [Rieger et al., 2010] or under some other condition imposed by the researcher [Twala, 2009]. In the multivariate case, past work mainly distinguishes between a local vs global [Garciarena and Santana, 2017] approach to value removal; the former ensures that every feature has the same proportion of missing values, while the latter considers the entire dataset for value deletion which does not ensure such a stratified missingness condition. The generation of synthetic missing data will be further described in the context of our methods.

2.3 Related Work

The concept of missing data in literature is addressed from different perspectives depending on the purpose of the research itself. While some studies approach missingness as a preprocessing step in their actual endeavour, other work focuses solely on the techniques used to do so. This dichotomy highlights different view points, depending whether or not missingness of data is the object of interest in a study. We specifically elaborate on past work that relates to how missingness is reported and handled when solving real-world problems; i.e., the application of the missing-indicator method in the context of specific domains. Moreover, we highlight the performance measurement methods applied in the context of missing data-handling techniques according to the current literature.

Following the first topic of interest, authors in [Malla et al., 2018] conducted an overview study of how missingness is addressed in the context of propensity score estimation; 167 articles were analysed in their research. Nearly 68% of these articles based their findings on assumptions that would only hold if data were MCAR. However, only one of these studies presented evidence for such a strong assumption. The remainder offered no explanation towards the reason data was missing nor which missing mechanism was at play. This observation led to biased results and skewed reported conclusions which posed a serious issue, especially given that the contexts of these studies were medical trials.

In other medicine-related domains scrutinised by us, contradicting evidence is reported with respect to the application of the missing-indicator method. Specifically, while authors in [Groenwold et al., 2012] state that *"the missing-indicator method is a valid method to handle missing baseline covariate data, irrespective of the mechanism of missingness"*, the work presented in [Van der Heijden et al., 2006] concluded that *"in multivariable diagnostic research complete case analysis and the use of the missing-indicator method should be avoided, even when data are MCAR"*. This discrepancy in the current literature is indicative of a real substantial problem that can only be addressed through further exposition of missing data as a subject of interest, by generating research that aims to offer general guidelines that practitioners may follow on how to handle missing data. In this sense, our contribution addresses whether a particularly common method of handling missing data – the missing-indicator method – should be used under the testable MCAR mechanism scenario.

Focusing on a different scope, studies such as [Amiri and Jensen, 2016] report on imputation techniques and their performance. In this work, the authors developed a novel imputation method to be applied on missing numerical data. They compared their method against frequently used imputation methods and reported the comparative performances yielded. The performance was measured as a function of the error between the imputed values and the original values. Despite being an intuitive approach, using this error as a performance measure for imputation is not adequate. This is thoroughly elaborated in [Van Buuren, 2018], where the distinction between predictive methods and imputation is established. The author ultimately asserts that imputation is not prediction and states that *"we cannot evaluate imputation methods by their ability to re-create the true data"*. In short, using regression error to measure performance leads to biased conclusions.

Studies such as [Garciarena and Santana, 2017] address this imputation performance issue by using classifier performance (downstream task) as a proxy for imputation adequacy. In such a study, different imputation techniques (both single and multiple) were used, in association with several classification algorithms across distinct datasets. However, the missing-indicator method was not encompassed within the experimental setup provided.

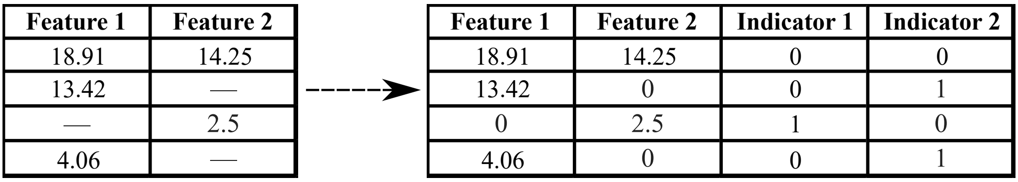
Taking these facts into consideration, we propose to comprehensively measure the impact of the missing-indicator method. We do so by using a downstream task as a viable proxy for missing data-handling performance under MCAR. Since MCAR is the only missing mechanism that can be tested against in real-world problems, we specify it as the base for our work; in this manner we ensure that the conditions under which our controlled experiments are performed can be statistically diagnosed in real-world scenarios.

2.4 Method

Here we provide the resources and methodology used. We begin with Section 2.4.1 by describing the application of the missing-indicator in the context of our work. In Section 2.4.2, we summarise the datasets we use in our experiments. Section 2.4.3 reports how missing data was synthesised. Lastly in Section 2.4.4, imputation and classifier methods are denoted.

2.4.1 Missing-Indicator

The missing-indicator method should not be regarded as an imputation method in itself. Rather, it can and should be viewed as an addition to any imputation being performed. In other words, regardless of what approach is used to fill in missing values – mean, median, regression-based imputations, etc. – the missing-indicator will always be applicable. The underlying concept of the missing-indicator method focuses on the encoding of missingness itself. In practice, this encoding can be regarded as the addition of a binary indicator variable. Concretely, our implementation of the missing-indicator method is as follows: every missing value is replaced with a placeholder value; then a second column is created for every feature with missing values. This new column holds values of either 0 or 1 representing the absence or presence of a missing value in the original feature, respectively (Fig. 2.1). This approach is derived from past literature, where every value $x_{i,j} \in X$ is replaced by the product of itself multiplied by $(1 - m_{i,j} \in M)$ [Bennett, 2001]. Our choice of a placeholder value of zero reflects also the consensus in methodological approaches applied by practitioners [Zhang, 2016].



Feature 1	Feature 2
18.91	14.25
13.42	—
—	2.5
4.06	—

Feature 1	Feature 2	Indicator 1	Indicator 2
18.91	14.25	0	0
13.42	0	0	1
0	2.5	1	0
4.06	0	0	1

Figure 2.1: **Imputation through missing-indicator.** The table to the left represents a 4-rows slice of some dataset with missing values in feature column 1 and feature column 2. The table to the right represents the yielded version of the previous table using the missing-indicator method.

2.4.2 Data

A total of 22 datasets were collected from an open-source dataset repository [Alcalá-Fdez et al., 2009], all of which were associated with a classification task.

Every dataset is comprised of a set of numerical features and a class column. These datasets are complete (i.e., no missing values) and vary significantly in sample size, dimensionality, and class balance. The definition of class balance follows.

Definition 2.1 – ***Class balance***

Class balance is the quantification of the difference between the number of samples pertaining to the positive and negative classes in a dataset.

A summary of the datasets can be seen in Table 2.2. The column *Class Balance* represents the ratio between the frequency of the minority class and the majority class: a value close to 0 indicates large class imbalance, while a value of 1 means perfect class balance.

Table 2.2: Summary of dataset characteristics

Dataset	#Rows	#Features	Class Balance
Appendicitis	106	7	0.25
Australian	690	14	0.80
Bands	365	19	0.59
Bupa	345	6	0.73
Coil2000	9822	85	0.06
Haberman	306	3	0.36
Heart	270	13	0.80
Hepatitis	80	19	0.19
Ionosphere	351	33	0.56
Magic	19020	10	0.54
Mammographic	830	5	0.94
Monk-2	432	6	0.90
Phoneme	5404	5	0.42
Pima	768	8	0.54
Ring	7400	20	0.98
Sonar	208	60	0.87
Spambase	4597	57	0.65
Spectfheart	267	44	0.26
Titanic	2201	3	0.48
Twonorm	7400	20	1.00
Wdbc	569	30	0.59
Wisconsin	683	9	0.54

We specify class balance as it plays a role when selecting the appropriate performance metric to measure classifier performance. Past literature states that given an imbalanced classification problem, the area under the precision-recall curve is more informative than the AUC [Saito and Rehmsmeier, 2015], which is equivalent to the average precision measure (AP). Thus, we usedAP to measure classifier performance, as well as specifying the minority class as the class to be modelled for every dataset. Given our aim at producing a comparative study, we address data and its characteristics in this segment rather than as part of our experimental setup.

2.4.3 Value Removal

To compare different imputation methods and their impacts on classifier performance for various missingness rates, artificially removal of values from the complete datasets was required. Following common use in literature, we removed 10%, 20%, 30%, 40%, and 50% of observations to measure how different percentages of missing values affect the impact of particular missing data-handling methods. We cap the missing proportions at 50% since higher values might damage the original dataset too much to extract meaningful results.

We used a multivariate local approach to generate missing values. In this manner, every feature had the same percentage of missing values for a given proportion of missingness. The removal followed a uniform distribution so that every value within a feature vector had the same probability of being removed. In practice, given some dataset subset with n observations, and p features, for a fixed missing proportion $q \in [0, 1]$, each feature vector had $\lceil q \times n \rceil$ observations removed. For clarification, we remark that a class label is not a feature.

We specify the notion *subset* because we did not apply missingness to the entire dataset at once. Rather, we first split the dataset into 10 equal segments and removed observations in each segment. These segments were used later on to compute classifier performances; i.e., they comprised our train-test splits. Should we have created our train-test splits a posteriori, then segments of the dataset could have had fallen under a non-homogeneous MCAR assumption, given the randomness of the splitting process. Thus, we ensured that all train and test segments used were under the same missing proportion conditions.

2.4.4 Imputers and Classifiers

Four simple and commonly used imputation methods were selected and implemented to serve as comparison against missing-indicator: mean (Mean), median (Median), linear regression (Linear), and extreme gradient boosting regression (XGBR). These methods were selected for their differing frameworks.

One important factor to take into consideration is which imputation framework to use: single or multiple imputation. We chose to implement the single imputation variant of each algorithm for our comparisons, rather than performing a multiple imputation implementation. The rational behind our decision is given below.

In multiple imputation [Carpenter and Kenward, 2012], several distinct imputed versions of the original missing dataset are generated. In practice, this framework makes use of any single imputation method (such as linear regression imputation) and adds to it a component of randomness (by bootstrapping, for example). This generates different complete variations of the same original missing-valued dataset. The resulting analysis output of each complete dataset is then combined (i.e., pooled). We chose single over multiple imputation because all compared imputation methods could be wrapped within a multiple imputation framework; we are only interested in comparing the imputation methods themselves, not the outcome of single vs multiple imputation.

While for mean and median imputations the missing values depend on the available values in the same column, for the regression-based imputers the values in a column are assumed to depend on the values of the same sample in other columns. Consequently, for a regression-based imputer, to impute values for column j, all other columns should be complete. This observation is based on the ones used in the literature [Bertsimas et al., 2017, Van Buuren, 2018]. In the first case, random values are generated to populate the initial incomplete dataset, whereas the latter uses mean imputation to do so. We refer to these initial imputation states as *warm starts* hereinafter. A warm start is defined as follows [Van Buuren, 2018].

Definition 2.2 – ***Warm start***

A warm start in prediction-based imputation is the initial step in which the dataset is made complete so as to enable learning of a model towards predicting the imputation values for the feature of interest.

In our case, these warm starts are the complete version of a missing-valued dataset, generated through mean imputation. They serve as a starting point for regression-based imputation, allowing for regressors to be trained by using a provisional complete matrix. After a regression model is fit, imputation can be performed, and classification models can be learned from the complete data.

The classification algorithms implemented were a k-nearest neighbours classifier (KNNC), a support vector machine classifier (SVMC) with a radial basis function kernel, and an extreme gradient boosting classifier (XGBC). We chose these algorithms as they cover the spectrum of the current state-of-the-art and their documented application in several domains.

2.5 Experiments

In this segment we address our experimentation. In Section 2.5.1 we describe how the methods mentioned previously were implemented. In Section 2.5.2, we present the resulting outputs. Lastly, we discuss our results in Section 2.5.3.

2.5.1 Experimental Setup

Our implementation was performed in *Python* using peer-reviewed libraries [Pedregosa et al., 2011, Chen and Guestrin, 2016, Oliphant, 2007], which are open-source. All programming objects were initialised using the default set of parameters supplied by each object's corresponding package, safe for random state parameters. For reproducibility purposes, a random seed value of 42 was set where appropriate (i.e., tree-based algorithms and sample selection during train-test splitting). We describe and illustrate how the aforementioned methods were applied within our experimental setup.

For each dataset, we split the entire dataset into 10 segments (folds) in a stratified manner as to ensure class balance across all folds as in the entire dataset. Through 10-fold CV we computed benchmark AP values per classifier. These benchmark values are derived from the original complete datasets and will serve to illustrate the differences across imputation methods.

In each fold, features had their values deleted according to our previously defined value-removal approach. In this manner, five distinct instances of the same fold were generated where each instance has a specific percentage of missing values. Within a dataset, for each missing proportion, we created 10 train-test sets. Each of these train-test sets was comprised of one distinct fold that served as the testing subset, and the joint set of the other remaining folds that served as the training subset. This setup of train-test splits was used to both apply the aforementioned imputations methods, as well as compute classifier performances.

Every configuration of splits per dataset for a specific missing proportion contained the same index instances. In other words, every imputation and classification procedure was always applied to the same subsets of the original dataset across the different algorithms to be compared. After structuring and creating all splits per dataset according to different missing proportions, we began the imputation processes.

For the missing-indicator, imputations required no distinction between train and test subsets: for every subset, missing values are converted to 0 while adding an extra dimension per feature, valued either 0 or 1 as previously described. While this method was applied without differentiating train or test instances, such was not the case for the remaining imputers.

Both regression-based imputation methods Linear and XGBR required warm starts to be initialised and applied. Since warm starts are yielded by Mean and given that Mean and Median applied similarly (although computing different statistics), we proceed to describe the imputation setup of both these methods.

Given a train-test split within our experimental setup, let X_{train} and X_{test} be the sets corresponding to the train and test portions of the split, respectively. Let $X_{train}^{obs} \in X_{train}$ and $X_{train}^{miss} \in X_{train}$ be disjoint sets representing observed and missing values, correspondingly, of the train portion of the train-test split. Conversely, let $X_{test}^{obs} \in X_{test}$ and $X_{test}^{miss} \in X_{test}$ be disjoint sets representing observed and missing values, respectively, of the test portion of the train-test split. For each feature indexed at $j \in \{1, ..., p\}$, mean and median statistics were computed from the set of observations $x_{:,j} \in X_{train}^{obs}$, for a total of p mean values and p median values per train-test split. We then replaced the missing values in both X_{train}^{miss} and X_{test}^{miss} with the mean or median of the respective feature.

The setup used to deploy both Linear and XGBR was the same. We started by considering any arbitrary train-test split and retrieving the corresponding imputed train-test set produced through Mean. These imputed train and test subsets served as the warm starts required to generate imputations through both regression-based methods. Each of these methods generated imputations based on the regression methods associated to them. For each train-test split, a total of p regression models were learned per method, and per feature.

Let R_j be an object representing the regression model to be used to impute over feature $j \in \{1, ..., p\}$. R_j had to be passed an initial set of values from which to learn. The values were a subset of the training warm start. This subset had the same width as the original corresponding dataset, but only contained the instances for which feature j has observations. R_j would then be trained on the subset in question to learn to model feature j from all other remaining features. Past the regression-training step, imputation was performed by R_j on both test and train segments of the train-test split.

After all imputation methods were applied to all train-test splits derived from each dataset for all missing proportions, classifier performance was computed. AP values were generated through 10-fold CV using the aforementioned train-test splits. The values were then averaged so that each classifier yields one value of AP per dataset per imputation method per missing proportion.

2.5.2 Results

We computed the mean performance over the 22 datasets, fixed on missing proportion, per imputer, for each classifier. Figs. 2.2, 2.3, and 2.4 illustrate the performance per imputers KNNC, SVMC, and XGBC, respectively.

Wilcoxon signed-rank tests [Wilcoxon, 1992] were applied to measure the statistical significance of the differences between missing-indicator and the remaining methods. The resulting p-values associated with KNNC, SVMC, and XGBC are shown in Table 2.3, Table 2.4, and Table 2.5, respectively.

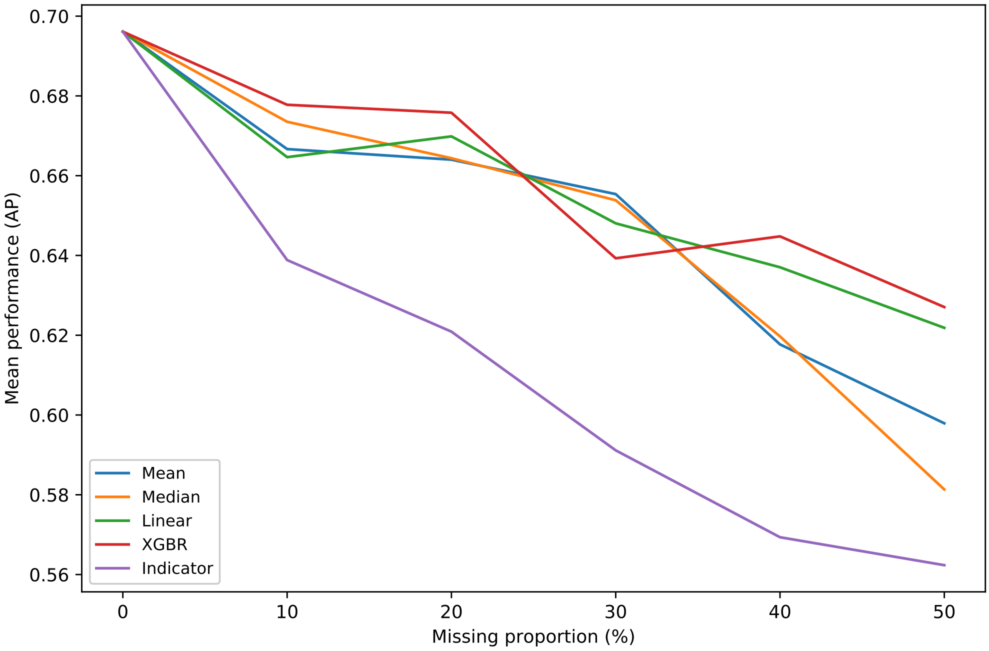


Figure 2.2: **KNNC mean performances across missing proportions.** Mean performances (vertical axis) across different proportions of missingness (horizontal axis) per five missing data-handling methods. Indicator refers to missing-indicator (purple).

Table 2.3: KNNC Wilcoxon p-values

	Mean	**Median**	**Linear**	**XGBR**
10%	0.030853	0.001549	0.020271	0.005506
20%	0.000779	0.006082	0.000136	0.001731
30%	0.002401	0.001549	0.002401	0.004981
40%	0.001932	0.001103	0.000259	0.000069
50%	0.094528	0.223429	0.015577	0.001549

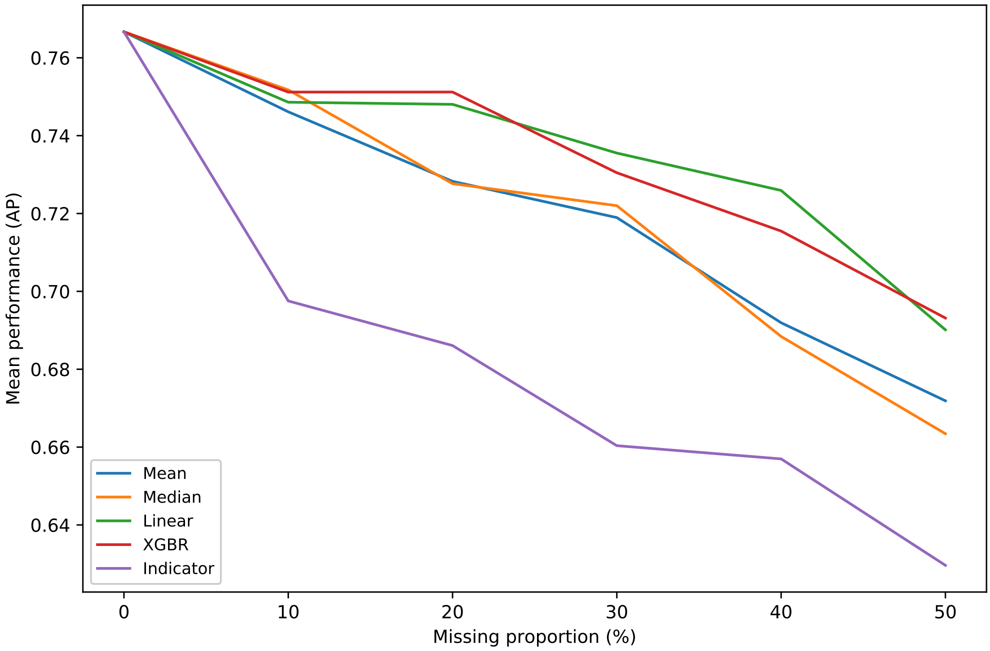


Figure 2.3: **SVMC mean performances across missing proportions.** Mean performances (vertical axis) across different proportions of missingness (horizontal axis) per five missing data-handling methods. Indicator refers to missing-indicator (purple).

Table 2.4: SVMC Wilcoxon p-values

	Mean	**Median**	**Linear**	**XGBR**
10%	0.026155	0.014239	0.004063	0.001237
20%	0.033462	0.030853	0.000136	0.000155
30%	0.001731	0.001932	0.000334	0.000615
40%	0.039249	0.013005	0.000483	0.002673
50%	0.014239	0.088298	0.000483	0.001237

Within each classifier setup fixed at missing proportion, the 22 average AP values computed using missing-indicator were compared against the remaining methods in a pairwise fashion. One p-value was yielded for each comparison of missing-indicator vs per imputation method fixed at a given missing proportion.

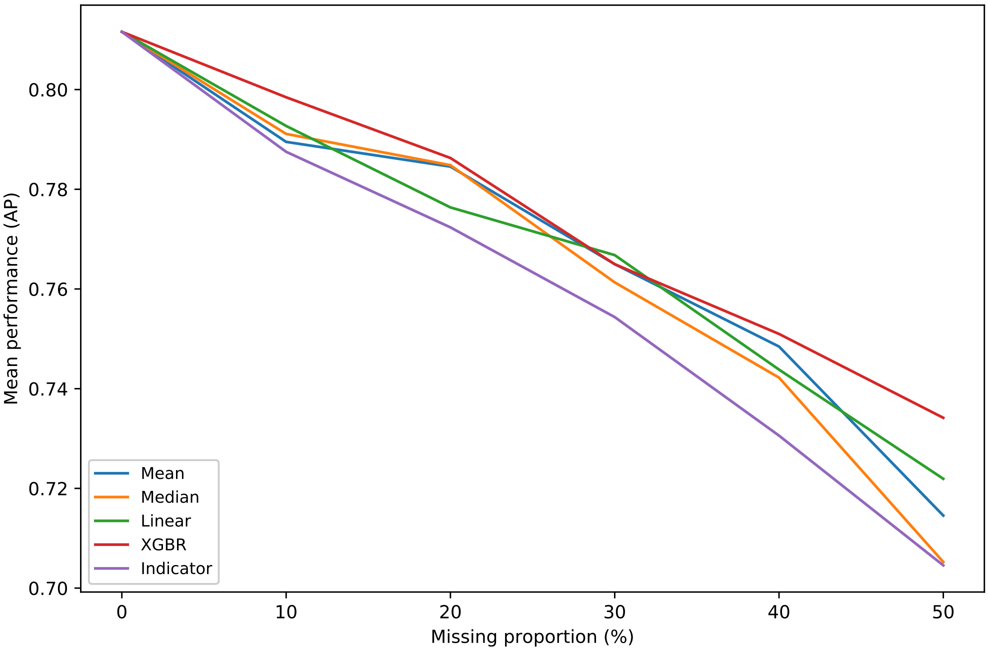


Figure 2.4: **XGBC mean performances across missing proportions.** Mean performances (vertical axis) across different proportions of missingness (horizontal axis) per five missing data-handling methods. Indicator refers to missing-indicator (purple).

Table 2.5: XGBC Wilcoxon p-values

	Mean	**Median**	**Linear**	**XGBR**
10%	0.807624	0.883846	0.987049	0.066608
20%	0.236019	0.445498	0.236019	0.082403
30%	0.445498	0.858282	0.020271	0.188557
40%	0.101106	0.426376	0.407742	0.139625
50%	0.262686	0.987049	0.066608	0.003302

2.5.3 Discussion

By definition —Eq. 2.1— the distribution of missing values under MCAR in a dataset is completely independent of any aspect of the dataset itself. By using the missing-indicator method, a new variable is introduced per missing-valued feature. The usage of the missing-indicator method generates a binary variable of which the distribution is also independent of any aspect of the data.

In other words, under the MCAR scenario this missing data-handling approach is adding a noise variable to the data which will make the classification task more difficult to solve.

Classifiers such as XGBC, which incorporate feature selection mechanisms, are less prone to be influenced by the added noise through the incorporation of missing-indicator variables (Fig. 2.4). However, should a classification algorithm be not so robust as assumed, then the overall classification performance is expected to drop. This is illustrated in Fig. 2.2 (KNNC) and Fig. 2.3 (SVMC), where an accentuated difference in average performance with missing-indicator vs all other imputers can be seen. The distinction is further denoted statistically by analysing Tables 2.3, 2.4, and 2.5. In the first two tables, nearly all p-values are under 0.01 (i.e., significant difference between missing-indicator and imputation methods). Table 2.5 suggests little disparity in performance between imputation and missing-indicator methods.

2.6 Chapter Conclusion

Handling missing data is a general problem encountered in most machine learning tasks. Different methods exist in the literature to address this problem, of which imputation and missing-indicator are the most predominant. Depending on underlying missing mechanism, the learner used, and the choice of missing data-handling method, the downstream task performance may vary.

When dealing with real-world data, a non-MCAR scenario is traditionally assumed and a viable option is to use the missing-indicator method. However, even with a negative MCAR test output, a false negative may still be possible which may jeopardise the performance of the downstream task. Accounting for this liability, it is necessary to attenuate the performance decrease derived from using the missing-indicator method under MCAR.

In this work, we have extensively assessed the performance of the missing-indicator approach under the testable MCAR missing mechanism towards a downstream classification task. We compared it to common imputation methods based on both statistical and machine-learning approaches. We computed classifier performances from three distinct algorithms applied to 22 datasets, each instanced with different proportions of missing values.

The comparative impact on classifier performance of each imputer was illustrated, and statistical significance tests were applied to further validate our findings. We observed that, as expected, the missing-indicator method systematically underperforms relative to all imputation methods. Yet, the negative impact of the missing-indicator method (compared to imputation methods) can be made negligible via adequate learner selection.

In conclusion, our research shows that the missing-indicator method is a viable option when handling real-world data, even if the missing mechanism is not correctly assessed, so long as a decision tree-base learner is used,concretely via gradient boosting. As a closing remark for upcoming research, we state that using real missing-valued datasets rather than ones with synthetically generated missingness might provide more realistic and robust results.

Chapter 3

Crosslier Detection

Finding anomalous entries is a difficult task with real-world consequences. Consider the following example. Transit of wasteful materials within the EU is highly regulated through a system of permits. Waste processing costs vary greatly depending on the waste category of a permit. Therefore, companies may have a financial incentive to allege transporting waste with erroneous categorisation (i.e., label-noisy samples). Our goal is to assist inspectors of the ILT in selecting potentially manipulated permits for further investigation. For this purpose, we introduce the concept of *crosslier*, of which the definition follows.

Definition 3.1 – ***Crosslier***

A crosslier is a sample of which (a) the category label is swapped and (b) a proportion of its features is more similarly valued to the features of the samples of the newly-swapped category.

To detect crossliers, we propose the EXPOSE method. Moreover, to facilitate the targeting of crossliers by inspector, we define the *crosslier diagram*.

Definition 3.2 – ***Crosslier diagram***

A crosslier diagram is a visualisation tool specifically designed for domain experts to easily assess crossliers.

We compare EXPOSE against traditional detection methods in various benchmark datasets. By evidencing the superior performance of our method in targeting these instances of interest, we provide an answer to RQ2(a): given data with label noise, how can noisy-samples be adequately detected?

The current chapter corresponds to the following publication:
Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2021). The eXPose approach to crosslier detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2312–2319. IEEE

3.1 Crossliers and Miscategorisation

Within the EU, economic proliferation and globalisation have resulted in a increase of transnational waste transportation. The nowadays established List of Waste provides EU member-states with waste categorisation, which promotes appropriate waste handling, particularly relevant for hazardous waste [European Commission, 2018a]. Since transportation of waste poses serious health and environmental risks, all movement of waste must be priorly noticed through a system of permits [European Commission, 2018b]. In the Netherlands, the entity responsible for permit compliance is the ILT. Inspectors must evaluate and determine whether a permit (a) is likely to be compliant and requires no further inspection, or (b) raises concern and requires investigation.

Since different waste categories are encompassed by specific regulations with dissimilar processing costs, companies may have an economic incentive to purposefully miscategorise their waste. Hence, targeting such cases is of utmost importance to the inspectors of the ILT. Given high volume and velocity of data, however, inspectors cannot adequately assess all permits. Therefore, automatic methods are required.

Under the current problem scenario, the usually most-effective supervised learning approaches to instance targeting [Choudhary and Gianey, 2017] are not applicable since no historical labels for misconduct are available. Unsupervised learning techniques are also not suited, given the unspecificity of the retrieved instances. Here we note that for anomaly detection methods, outlyingness alone does not translate to the desired targets, and we further mention the difficulty of detecting samples in high-dimensional data [Venkatesh and Anuradha, 2019].

With respect to data-quality assurance techniques, we remark that they mostly depend on variable distribution assumptions and concentrate on random errors [Liu et al., 2016]. We focus on instances in which the category label and category-correlated feature values have been altered. In other words, our goal is to pinpoint samples with *non-random* changes in feature values which mask the true underlying category label.

To address the current problem of manipulation, we propose the following three contributions:

1. the concept of a *crosslier*: a deviating instance resulting from potentially intentional category manipulation;
2. the EXPOSE method for crosslier detection, by computing the crosslier score of a sample given its category;
3. the *crosslier diagram*: a visualisation tool which allows easy assessment of crossliers.

Albeit motivated by a waste transportation problem, our proposed contributions are intrinsically domain-agnostic and therefore applicable to other fields. Within a dataset with category labels, a crosslier is an instance of which the combination of (1) its set of feature values and (2) the category label are disharmonious.

We consider a crosslier to be a *special* case of an outlier defined as follows.

Definition 3.3 – ***Outlier***

An outlier is a sample of which the feature values differ significantly from those of the other samples.

By special case, we mean that crossliers are outlying instances with *specific* characteristics. More precisely, a crosslier is a specific outlier with some connection regarding a category label; that is, it is a sample of a category which lies *across* other categories.

For completion, here we remark that the terms (a) crosslier and (b) outlier are both a form of (c) anomaly, which is defined as follows.

Definition 3.4 – ***Anomaly***

An anomaly is a sample which, given its features, class label, domain knowledge, or any combination of the three, is significantly different from the remainder of the samples. It is used to broadly refer to a data point which stands out from the dataset.

The relationship between the three terms is depicted in Fig. 3.1. As shown, all crossliers are outliers and all outliers are anomalies, but not all anomalies are outliers and not all outliers are crossliers.

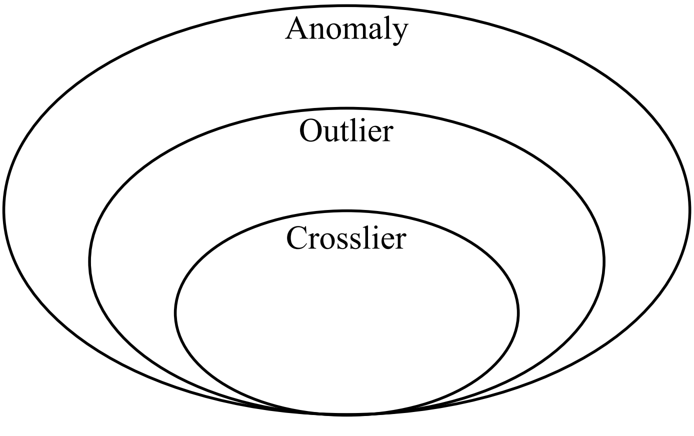


Figure 3.1: **Anomaly, outlier, and crosslier.** Diagram depicting the relationship between the three terms.

The chapter structure follows: Section 3.2 states our problem formally; Section 3.3 discusses past work related to ours; Section 3.4 elaborates our approach in detail; Section 3.5 describes ours experimental setup; Section 3.6 refers to our results; Section 3.7 discusses our method; and Section 3.8 concludes this work and suggests future research directions.

3.2 Problem Description

Given a category-labelled dataset, we defined (in Definition 3.1) a *crosslier* as a sample of which (a) the category label is swapped and (b) a proportion of its features is more similarly valued to the features of samples of the newly-swapped category. To put it simply, we assume that feature values might have been manipulated to mask the true category label.

To detect crossliers, we propose *crosslyingness* as a rankable property expressed as a function, in which the instance with the highest crosslyingness with respect to a category is the most likely crosslier.

Definition 3.5 – ***Crosslyingness***

Crosslyingness is a rankable property indicative of the degree to which a sample is considered a crosslier.

Accordingly, either (1) crossliers fall within the cluster of some other category, or (2) crossliers lie across other categories. To illustrate, we present Fig. 3.2; four different categories $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ are denoted, with crossliers as $\mathcal{A}^*$, $\mathcal{B}^*$, $\mathcal{C}^*$, and $\mathcal{D}^*$.

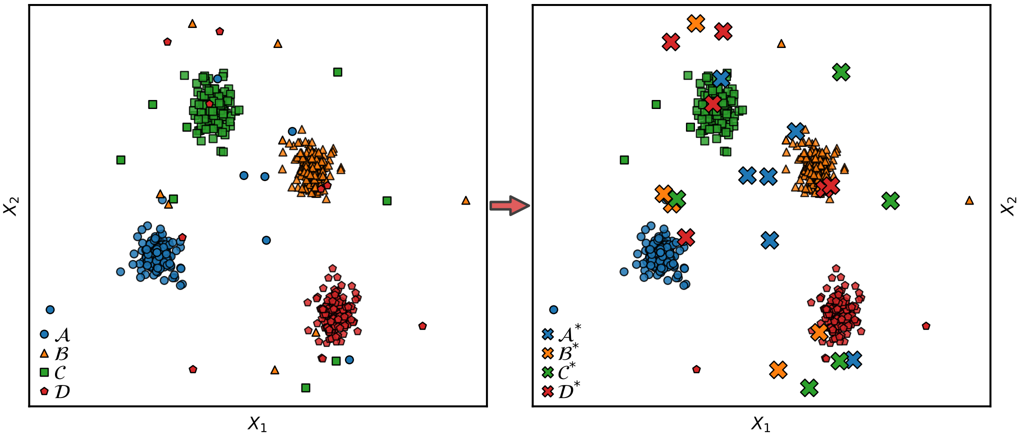


Figure 3.2: **Crosslier detection.** Samples with features X_1 and X_2, pertaining to either category $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, or $\mathcal{D}$ (left). Crossliers are marked as crosses (right).

Formally, let $\mathcal{D}$ be a distribution of random variables $(X, Z) \in \mathcal{X} \times \mathcal{Z}$, where $\mathcal{X} \subseteq \mathbb{R}^m$, $\mathcal{Z} = \{z_1, z_2, \dots, z_q\}$, and $z \in \mathcal{Z}$ is one of the q different category labels. Let also $(x_1, z_1), \dots, (x_n, z_n)$ represent the samples drawn from $\mathcal{D}$. Our goal is to find, for each unique category $z \in \mathcal{Z}$, a function $f_z(x)$ which scores the crosslyingness of $x_i \in X$ with $z_i = z$.

3.3 Related Work

In this section, we provide a brief overview of three techniques typically used to address anomaly detection problems. In this sense, we consider a crosslier to be a particular type of data anomaly, with specific characteristics as described in the previous sections.

We report on previous work which applied supervised and semi-supervised learning techniques (Section 3.3.1), unsupervised learning methods (Section 3.3.2), and data quality assurance techniques (Section 3.3.3). We further disclose their non-applicability to our scenario.

3.3.1 Supervised and Semi-supervised Learning

In the presence of labels indicative of previously-recognised noncompliance, the problem can be approached as a supervised learning task. Three examples are: (1) detecting insurance fraud [Subudhi and Panigrahi, 2020]; (2) exposing deceitful telecommunication users [Li et al., 2018]; and (3) identifying irregular heart beat patterns [Vollmer et al., 2017]. The choice of algorithm is rather diverse. We mention three of them: (1) SVMC [George and Vidyapeetham, 2012]; (2) multilayer perceptron [Mulongo et al., 2020]; and (3) random forest [Alazzam et al., 2019].

For the case where both labelled and unlabelled instances are available, a semi-supervised learning approach is suitable [Chapelle et al., 2006]. This framework can, as an example, make use of clustering algorithms assuming that data points within the same cluster probably share the same label [Xiang and Min, 2010]. Another approach to improve on the selection of inspection targets is to consider the unlabelled instances as pertaining to the negative class (i.e., the class which is not of interest) [Jacobusse and Veenman, 2016]. Here, the assumption is that the incidence of inspection targets within the unlabelled data is small enough as to be made negligible towards learning. Yet, our data does not possess target labels, making these techniques inapplicable.

3.3.2 Unsupervised Learning

A straightforward alternative is to find deviating cases through *anomaly detection* techniques using unsupervised methods. The assumption is that the most probable samples to target are the ones that differ in an extreme way from all others in their category (i.e., outliers). Such techniques have been applied to system intrusion detection [Zanero and Savaresi, 2004], maritime traffic anomaly flagging [Vespe et al., 2012], and image curation [Liu et al., 2014], amongst others. Four examples of the successful algorithms used are: (1) isolation forest (IF) [Liu et al., 2012]; (2) local outlier factor (LOF) [Breunig et al., 2000]; (3) nearest-neighbour [Amer and Goldstein, 2012]; and (4) k-means clustering [Muniyandi et al., 2012].

There are at least three intrinsic obstacles with unsupervised methods. The first obstacle is their dependency on distance metrics (Minkowski measures) to define outlyingness, which makes them sensitive to feature scaling. The second obstacle arises when dealing with high-dimensional data [Liu et al., 2017], particularly when attempting to estimate densities empirically [Santos et al., 2019a]. The third obstacle is that, through manipulation of only a proportion of features — as per the problem description (see Section 3.2)— target samples (crossliers) may not stand out. To illustrate, we present Fig. 3.3.

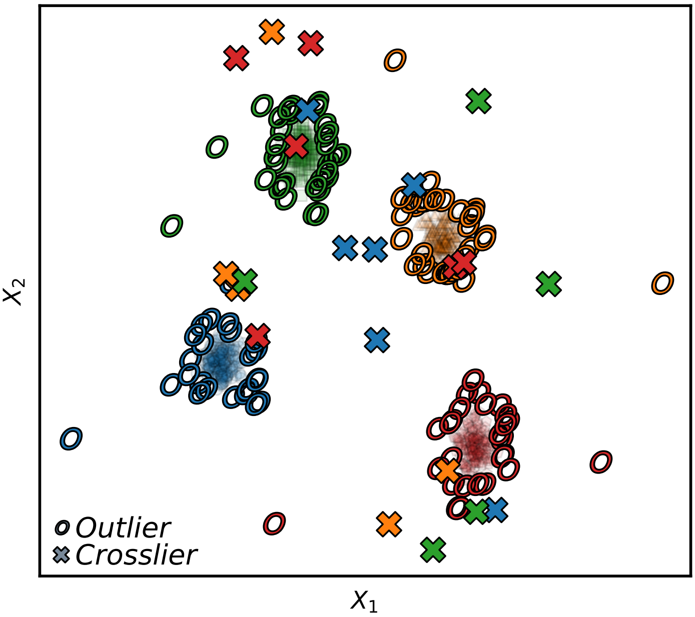


Figure 3.3: **Distinction between outlier and crosslier.** Four-category example from Fig. 3.2. Crossliers are marked as crosses and outliers are denoted as circles. Transparency values for data clusters have been raised for visualisation.

Fig. 3.3 builds on the example in Fig. 3.2 by applying the IF algorithm as per [Liu et al., 2008]. Here we see how data points flagged as outliers *do not* represent the target crossliers; hence, we should consider the distribution of categories when marking instances as crosslying. The issue with using traditional anomaly detection methods towards finding crossliers is evidently illustrated: most flagged instances are arguably not outlying with respect to their clusters. The insensitivity shown makes anomaly detection methods precarious to address our problem. Ultimately, not all outliers are crossliers since not all of them possess the specific category-related characteristics we seek.

3.3.3 Data Quality Assurance

By considering an outlier to be anomalous, and therefore an inspection candidate, one could argue that the abnormal values by which outlyingness is attributed can be caused by erroneous data entries on the permit category. Here, data quality assurance techniques can be used for detection [Bonner et al., 2015]. Typical methods involve, for example, assumptions over feature distributions [Mariet et al., 2016] and cross-referencing datasets for dependency-matching or constraint-mining [Rekatsinas et al., 2017, Chu et al., 2013]. Our scenario does not allow for reliable cross-dataset linkage due to the lack of entity identifiers. Furthermore, despite the existence and usage of both univariate and multivariate constraints, the constraints are not generated with respect to an ulterior task. In other words, the assumptions over feature distributions need not hold for the category distributions we are interested in.

In summary, the current literature is ill-equipped to adequately address our issue of discriminating towards crosslying instances, which translate to permits of interest to inspectors.

3.4 The EXPOSE Method

Here, we detail the proposed EXPOSE for the detection of crossliers. As defined in Section 3.2, the aim is to find a function $f_z(x)$ that determines the crosslier score of sample $x \in X$ with category label z. The EXPOSE method is data-driven in the sense that it uses a learning function to obtain the scores for a dataset with category labels. Since the whole dataset is category-labelled by definition, all samples can obtain a crosslier score. We follow a supervised learning approach, where the crosslying score is determined per category on a left out part in order to obtain an independent score. As a result, we need to optimise several learners as in a CV setup. Therefore, these learned functions must be calibrated to make the scores comparable among each other.

Below, we first describe the setup to obtain the learners in a supervised way. We then elaborate on the model selection and model calibration steps per data subset based on CV. The learners collectively yield the overall crosslier score function. We finalise the method section with the crosslier diagram, a tool to visualise crosslier scores and pinpoint suspect samples.

3.4.1 Classification Setup

Consider the distribution $\mathcal{D}$ defined in Section 3.2. For a fixed category z, $(x_1, y_1), \ldots, (x_n, y_n)$ are samples of $\mathcal{D}$ in which

$$y_i = \begin{cases} 1, \text{ if } z_i = z \\ 0, \text{ otherwise} \end{cases} \tag{3.1}$$

Given $\mathcal{D}$ and a loss function $\mathcal{L}$, the task of the learner is to find a function $f \in \mathcal{F}$ through empirical risk minimisation [Vapnik, 2013]:

$$\underset{f \in \mathcal{F}}{\arg\min}\, \hat{\mathcal{R}}_{\mathcal{D},\mathcal{L},f} \tag{3.2}$$

where

$$\hat{\mathcal{R}}_{\mathcal{D},\mathcal{L},f} = \frac{1}{n} \cdot \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i) \tag{3.3}$$

Depending on the chosen learner, the curse of dimensionality is addressed by incorporating either regularisation, feature selection, or both protocols in the learning task [Sharma et al., 2017]. These protocols also alleviate overfitting and promote classifier robustness by reducing the complexity of the final model [Gupta et al., 2016].

All regularisation parameters given prior to the learning task can be optimally retrieved through hyperparameter optimisation [Claesen and De Moor, 2015, Bergstra and Bengio, 2012]. The learners to be applied within a specific problem can also be optimally selected.

3.4.2 Model Selection

A model is selected based on classification performance. For each candidate learner that is applicable to a problem and their respective hyperparameters, the estimated classification performance is measured in terms of AUC through CV [Flach, 2016]. The choice of CV strategy is dependent on $\mathcal{D}$, as the appropriate number of folds and splitting strategy relate to $\mathcal{Z}$ and the respective $P(y)$, as well as sample size n. Model calibration is also subject to the CV strategy, detailed further.

Formally, consider the dataset D, with distribution $\mathcal{D}$. For a given $k \in \{1, 2, \ldots, K\}$, $K > 1$, let test set D_k^{ts} and training set D_k^{tr} be independent and identically distributed subsets of D such that

$$\bigcap_{k=1}^{K} D_k^{ts} = \emptyset, \bigcup_{k=1}^{K} D_k^{ts} = D, \text{and } D_k^{tr} = D \setminus D_k^{ts} \tag{3.4}$$

Fixing on k, we define test and training sets D_ℓ^{ts} and D_ℓ^{tr}, respectively, as independent and identically distributed subsets of D_k^{tr}, for $\ell \in \{1, 2, \ldots, L\}$ and $L > 1$, such that

$$\bigcap_{\ell=1}^{L} D_\ell^{ts} = \emptyset, \bigcup_{\ell=1}^{L} D_\ell^{ts} = D_k^{tr}, \text{and } D_\ell^{tr} = D_k^{tr} \setminus D_\ell^{ts} \tag{3.5}$$

Given D and sets of learners $\{\Psi_1, \Psi_2, \ldots, \Psi_r\}$ with hyperparameters $\{\phi_1, \phi_2, \ldots, \phi_p\}$, the final model is selected by maximising the estimated AUC with K and L folds, comprised of learner $^*\Psi$ and hyperparameters $\phi_k \in \{\phi_1, \phi_2, \ldots, \phi_K\}$. AUC is directly linked to crosslyingness, as detailed ahead.

Learner $^*\Psi$ and hyperparameters ϕ_k are used to generate the crosslier scores. Since eXPOSE generates crosslier scores from a collection of models learned on independent data subsets to avoid overfitting, the output of each model is not comparable across models. We enforce model comparability through model calibration.

3.4.3 Crosslier Score

To transform the output of uncalibrated models into a calibrated output, Platt scaling [Platt et al., 1999] is used. The original output $\hat{y}$ of a learned model given input x thus becomes the estimated posterior probability $\hat{P}(y|x)$. Given z, the crosslier score function f_z is defined as the information content [Jones, 1979] of a sample x from category z:

$$f_z(x) = -\log_2 \hat{P}(y|x) \tag{3.6}$$

The choice of $-\log_2$ translates to: (1) the score difference between samples with low and high posterior probabilities are augmented; and (2) scores are easily interpretable, in which a posterior 1 returns a score 0, and a posterior 0.5 returns 1. Heuristically, samples with crosslier score greater than 1 can be considered crossliers and are rankable by crosslyingness according to their respective crosslier scores. The estimated AUC model performance relates to the crosslier scores. By definition, poor-performing models output calibrated posterior probabilities close to 0.5. Therefore, the crosslier scores will lie close to 1 for all samples. With high AUC models, the range of crosslier scores is allowed to widen.

Formally, let x_k and y_k represent the variable values of samples $(x, y) \in D_k^{ts}$ for a given k. The estimated posterior is then given as

$$\hat{P}(y|x) = \bigcup_{k=1}^{K} \hat{P}(y_k|x_k) \tag{3.7}$$

in which,

$$\hat{P}(y_k|x_k) = \frac{1}{L} \cdot \sum_{\ell=1}^{L} [f_\ell^k({}^*_\phi\Psi_k^{tr}(x_k))] \tag{3.8}$$

where ${}^*_\phi\Psi_k^{tr}(x_k)$ is the output of ${}^*\Psi$ learned on $(x, y) \in D_k^{tr}$ with hyperparameters ϕ_k, given input x_k, and f_ℓ^k is the sigmoid function with parameters α^* and β^*

$$f_\ell^k(u) = \frac{1}{1 + e^{-(\alpha^* + \beta^* \cdot u)}} \tag{3.9}$$

in which

$$\alpha^*, \beta^* = \arg\min_{\alpha,\beta} - \sum_{(x,y) \in D_\ell^{ts}} [\mu \cdot log(p) + (1 - \mu) \cdot log(1 - p)] \tag{3.10}$$

where

$$\mu = \begin{cases} \dfrac{(\sum_{y \in D_\ell^{ts}} y) + 1}{(\sum_{y \in D_\ell^{ts}} y) + 2}, \text{ if } y = 1 \\ (|D_\ell^{ts}| - (\sum_{y \in D_\ell^{ts}} y) + 2)^{-1}, \text{ otherwise} \end{cases} \tag{3.11}$$

and

$$p = \frac{1}{1 + e^{-(\alpha + \beta \cdot {}^*_\phi\Psi_\ell^{tr}(x))}} \tag{3.12}$$

In Eq. 3.12, ${}^*_\phi\Psi_\ell^{tr}(x)$ is the output of ${}^*\Psi$ learned on $(x, y) \in D_\ell^{tr}$ with hyperparameters ϕ_k, given input $x \in D_\ell^{ts}$.

3.4.4 Crosslier Diagram

At the basis of the crosslier diagram (see Definition 3.2) lies an interactive tool which discriminates individual samples based on their crosslier score. Existing tools such as box, swarm, and violin plots were not suited since: (1) box plots do not present all samples that might be relevant crossliers; (2) swarm plots do not function well for a large number of samples; and (3) violin plots do not exhibit any samples in their output.

The diagram is a mapping of the output of $f_z(x)$ onto a horizontal axis where x are samples of category z. To each plotted sample we add a Gaussian-generated vertical value so that even if two or more samples have the same crosslier score they do not entirely overlap.

Finally, the crosslier diagram can display related domain-specific information of a sample by hovering over it. In the context of real-world transportation data, we present the crosslier diagram (Fig. 3.4) in the upcoming Section 3.6 as part of our experimental results.

3.5 Experiments

In this section, we describe our experiments. Two setups are considered, viz. (a) waste transportation setup and (b) benchmark setup. Within the first setup, EXPOSE is applied to the waste permit dataset (Section 3.5.1). In the second setup, we compare our method to anomaly detection methods in a controlled environment (Section 3.5.2). The resources described in this section are made available online [Pereira Barata, 2020].

3.5.1 Waste Transportation Setup

In this section, we discuss: (1) data; (2) learners; and (3) selection and calibration of the best model.

Data

The dataset was generated and provided by the ILT. It represents solicitations of waste transportation events across Europe (2009–2015), encompassing a total of $876,311$ waste transportations. Each row represents an individual transportation event. Several rows are linked by a permit identifier, where permits are the units of interest to inspectors of the ILT. We followed an aggregation strategy with respect to permit identifiers. The aggregation process produced $11,740$ instances, each with a waste category (out of 20 total different waste categories) and 49 variables which were a mixture of numerical and nominal features.

Learners

We experimented with (a) linear and (b) non-linear learners to find the best performing model for each waste category. First, an elastic net-regularised logistic regression (LR) learner was deployed, with hyperparameters λ and ϵ referring to the regularisation coefficient, and the ratio of $L1$ to $L2$-regularisation, respectively. Besides its broad usage and proven efficacy [Rosario, 2004, Wang, 2005, Mok et al., 2010], advantages of this learner are, for example: its calibrated output probabilities (hence, not requiring any further calibration); and its resilience to overfitting given low complexity and regularisation [Kleinbaum and Klein, 2010].

Second, a non-linear gradient boosted tree framework (XGBC) was considered [Friedman, 2001], with 100 additive trees. Each tree was allowed a maximum depth of 3 with regularisation parameter $\lambda = 1$. This learner is widely accepted as a state-of-the-art solution to supervised problems [Pafka, 2019] in terms of scalability, robustness to noisy samples, and classification performance.

Selection and calibration

To select and calibrate the best model, we applied nested-CV in a stratified manner [Stone, 1974] with $K = 10$, $L = 10$ as described in Section 3.4. Stratification is selected to ensure that each category is represented in each fold with the same relative frequency as in the full dataset. A grid-search [Chan and Treleaven, 2015] was applied to find the optimal set of LR regularisation parameters λ and ϵ. Each parameter was set to one of 21 distinct values, in ranges $[10^{-3}, 10^{3}]$ logarithmic and $[0, 1]$ linear, respectively, for a total of 441 sets of candidate hyperparameters. Since XGBC is relatively insensitive to hyperparameter changes, as shown in the experimental results of [Xia et al., 2017], we did not perform hyperparameter optimisation for this classifier. The best model for each category was used to generate the crosslier scores and crosslier diagrams (Section 3.6).

3.5.2 Benchmark Setup

In this section, we discuss: (1) data; (2) preprocessing; (3) crosslier synthesis; and (4) evaluation.

Data

Twenty binary classification datasets were retrieved from *openML*: an open, organised, and online ecosystem for machine learning [Vanschoren et al., 2014]. They are real-world datasets from different domains. Target classes were treated as the categories $\mathcal{Z}$. Table 3.1 summarises each dataset with identifier ID, n instances, and m features of which u are numeric. The datasets were chosen such that n, m, and u are heterogeneous across datasets.

Preprocessing

Numeric features values were scaled to a $[0, 1]$ range to accommodate feature scale-sensitive methods. Non-numeric features were $\{0, 1\}$-binarised per unique value.

Table 3.1: Datasets retrieved for crosslier simulations.

ID	n	m	u	ID	n	m	u
446	200	7	6	40705	959	44	42
40	208	60	60	31	1000	20	7
1495	250	6	0	1494	1055	41	41
53	270	13	13	40706	1124	10	0
40710	302	14	5	1462	1372	4	4
59	351	34	34	1504	1941	33	33
40690	512	9	0	1487	2534	72	72
1063	522	21	21	1485	2600	500	500
335	554	6	0	41143	2984	144	8
1510	569	30	30	41144	3140	259	259

Crosslier synthesis

To simulate a real-world scenario, crossliers were synthesised by replacing category labels and feature values. Different proportions of both label and feature manipulation were considered extensively. The proportion of label-swapped samples for each category per dataset was $\rho_y \in \{.01, .05, .1, .15, .2, .25, .3, .35, .4\}$. To recreate the scenario in which feature values are manipulated to simulate another category, samples which were label-swapped had a proportion of their feature values replaced. The proportion of randomly-selected features to have their values replaced was $\rho_x \in \{0, .05, .1, .15, .2, .25, .3, .35, .4\}$.

Replacement values were drawn from univariate distributions with parameters estimated from the features of the category being mimicked, modelled as either: (a) the normal distribution $\mathcal{N}(\hat{\mu}, \hat{\sigma})$ for numeric features, where $\hat{\mu}$ is the estimated mean and $\hat{\sigma}$ is the estimated standard deviation; or (b) the multinomial distribution with estimated event probabilities $\{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_\pi\}$ where π is the number of unique feature values, otherwise. Crossliers were generated 10 times with different random initialisation seeds for all datasets per configuration (ρ_x, ρ_y) to account for randomness. Both categories per dataset were corrupted with crossliers before any method was applied.

Methods

The EXPOSE method was compared to two well-established anomaly detection methods: LOF and IF, mentioned in Section 3.3.2. The previously-established methods were not designed to detect crossliers.

To promote a reasonable comparison, EXPOSE was applied with a single set of learner and hyperparameters and no optimised model selection was performed. The model selected was a tree-based gradient boost learner and default hyperparameters of 100 trees of maximum depth 3 with regularisation $\lambda = 1$ [Chen and Guestrin, 2016]; calibration values K and L were set to 10. LOF neighbourhood size was set to 20 and IF number of trees was set to 100.

Evaluation

The crosslier scores of EXPOSE were generated as in Section 3.4; the anomaly scores of the anomaly detection methods were generated category-wise. For each category, crosslier detection performance was measured in AP [Liu and Özsu, 2009], a common measure in anomaly detection assessment [Xu et al., 2018]. Accordingly, the targets are the crossliers in each category. The performance of both categories in each configuration (ρ_x, ρ_y) were jointly averaged per dataset, and across initialisation seeds.

3.6 Results

Here, we present findings relative to both experimental setups: (*a*) EXPOSE applied to the real-world scenario of waste transportation in the inspection domain; and (*b*) EXPOSE compared to other anomaly detection methods in a controlled environment with benchmark datasets.

3.6.1 Waste Transportation

When applied to the waste transportation data, we show firstly the estimated AUC performances yielded by both candidate models LR and XGBC. The next step was presenting the crosslier diagrams of waste categories to the inspectors for assessment. Waste category 4 (waste from textile industries) was not shown due to insufficient number of instances.

Model performance and selection

Table 3.2 shows the estimated AUC performances and measured standard deviations yielded during the model selection step of EXPOSE, which were used to select the best model per category for crosslier detection. Values in bold indicate the highest performance per category of which the model was chosen.

Table 3.2: Model performances per waste category.

Category	LR	XGBC
1	0.983 ± 0.008	$\mathbf{0.985 \pm 0.010}$
2	0.868 ± 0.044	$\mathbf{0.919 \pm 0.037}$
3	0.868 ± 0.020	$\mathbf{0.908 \pm 0.027}$
—	—	—
5	0.672 ± 0.092	$\mathbf{0.755 \pm 0.082}$
6	0.740 ± 0.038	$\mathbf{0.794 \pm 0.037}$
7	0.776 ± 0.016	$\mathbf{0.821 \pm 0.015}$
8	0.798 ± 0.026	$\mathbf{0.856 \pm 0.025}$
9	0.867 ± 0.047	$\mathbf{0.915 \pm 0.047}$
10	0.737 ± 0.032	$\mathbf{0.788 \pm 0.035}$
11	0.815 ± 0.021	$\mathbf{0.896 \pm 0.016}$
12	0.860 ± 0.032	$\mathbf{0.897 \pm 0.031}$
13	0.609 ± 0.063	$\mathbf{0.720 \pm 0.062}$
14	0.776 ± 0.034	$\mathbf{0.817 \pm 0.024}$
15	0.841 ± 0.019	$\mathbf{0.883 \pm 0.016}$
16	0.695 ± 0.016	$\mathbf{0.753 \pm 0.019}$
17	0.845 ± 0.023	$\mathbf{0.889 \pm 0.022}$
18	0.894 ± 0.015	$\mathbf{0.921 \pm 0.015}$
19	0.806 ± 0.014	$\mathbf{0.851 \pm 0.013}$
20	0.719 ± 0.024	$\mathbf{0.779 \pm 0.027}$

XGBC provided the best performance for all categories and was selected to generate the crosslier diagrams. For clarity, AUC does not measure the performance of crosslier detection since no crosslier labels exist in this real-world problem.

Crosslier diagrams

In Fig. 3.4 the crosslier diagrams with scores generated by the selected model XGBC are shown. For demonstration purposes, we show crosslier diagrams of four waste categories: (1) exploration and treatment of minerals; (2) agriculture, food preparation, and processing; (9) waste from photography industry; and (18) human or animal healthcare. In addition, the interactive aspect of the diagram is represented for a sample of waste category 9, in which its permit identifier (ID 4358) and crosslier score (1.41) are shown.

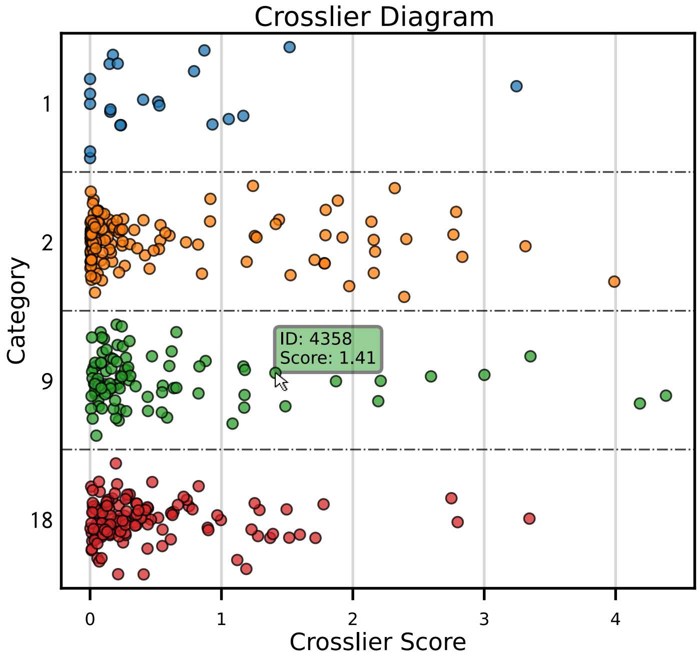


Figure 3.4: **Crosslier diagrams of four waste categories.** Hovering over an instance highlights its identifier (4358) and crosslier score (1.41).

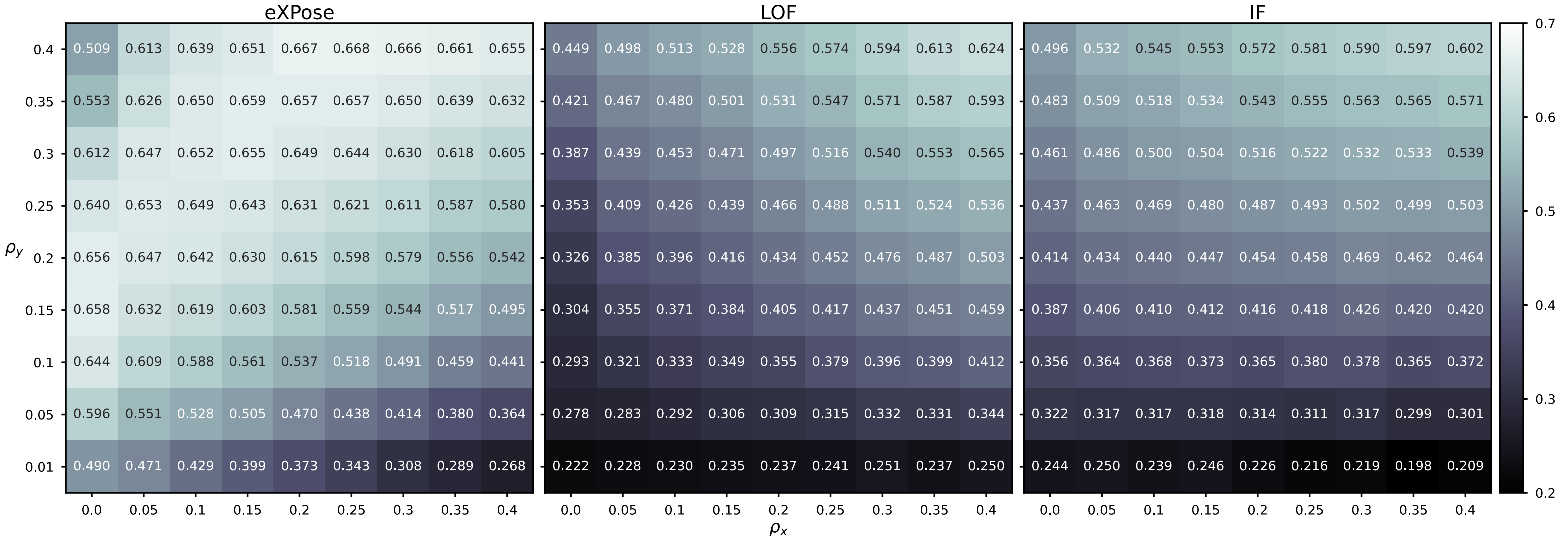
Inspection domain

The inspectors of ILT were provided with the crosslier diagrams. They analysed the permit cases across waste categories according to the given crosslier scores. Their assessment was that the authenticity of most of the high-scoring permits was sufficiently doubtful and that further investigation was necessary to establish compliance. All in all, the crosslier diagram was considered a valuable expansion of their tool set, especially when compared to spreadsheet analysis.

3.6.2 Benchmark

The outcome of our experiments with respect to controlled crosslier detection is to be seen in Fig. 3.5. We present the results for the three methods: EXPOSE, LOF, and IF. Fig. 3.5 shows the mean (AP) across 20 datasets, for 81 configurations of (ρ_x, ρ_y), each with 10 random initialisations of crosslier synthesis.

Lighter (darker) cell tones indicate higher (lower) values of performance. Each number indicates the yielded AP performance for each (ρ_x, ρ_y) configuration with which we experimented. For every possible setting (i.e., heatmap cell), EXPOSE yielded a higher mean performance than any of the other methods. The differences in performance diminish as both ρ_x and ρ_y increase.



eXPose

ρ_y \ ρ_x	0.0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4
0.4	0.509	0.613	0.639	0.651	0.667	0.668	0.666	0.661	0.655
0.35	0.553	0.626	0.650	0.659	0.657	0.657	0.650	0.639	0.632
0.3	0.612	0.647	0.652	0.655	0.649	0.644	0.630	0.618	0.605
0.25	0.640	0.653	0.649	0.643	0.631	0.621	0.611	0.587	0.580
0.2	0.656	0.647	0.642	0.630	0.615	0.598	0.579	0.556	0.542
0.15	0.658	0.632	0.619	0.603	0.581	0.559	0.544	0.517	0.495
0.1	0.644	0.609	0.588	0.561	0.537	0.518	0.491	0.459	0.441
0.05	0.596	0.551	0.528	0.505	0.470	0.438	0.414	0.380	0.364
0.01	0.490	0.471	0.429	0.399	0.373	0.343	0.308	0.289	0.268

LOF

ρ_y \ ρ_x	0.0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4
0.4	0.449	0.498	0.513	0.528	0.556	0.574	0.594	0.613	0.624
0.35	0.421	0.467	0.480	0.501	0.531	0.547	0.571	0.587	0.593
0.3	0.387	0.439	0.453	0.471	0.497	0.516	0.540	0.553	0.565
0.25	0.353	0.409	0.426	0.439	0.466	0.488	0.511	0.524	0.536
0.2	0.326	0.385	0.396	0.416	0.434	0.452	0.476	0.487	0.503
0.15	0.304	0.355	0.371	0.384	0.405	0.417	0.437	0.451	0.459
0.1	0.293	0.321	0.333	0.349	0.355	0.379	0.396	0.399	0.412
0.05	0.278	0.283	0.292	0.306	0.309	0.315	0.332	0.331	0.344
0.01	0.222	0.228	0.230	0.235	0.237	0.241	0.251	0.237	0.250

IF

ρ_y \ ρ_x	0.0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4
0.4	0.496	0.532	0.545	0.553	0.572	0.581	0.590	0.597	0.602
0.35	0.483	0.509	0.518	0.534	0.543	0.555	0.563	0.565	0.571
0.3	0.461	0.486	0.500	0.504	0.516	0.522	0.532	0.533	0.539
0.25	0.437	0.463	0.469	0.480	0.487	0.493	0.502	0.499	0.503
0.2	0.414	0.434	0.440	0.447	0.454	0.458	0.469	0.462	0.464
0.15	0.387	0.406	0.410	0.412	0.416	0.418	0.426	0.420	0.420
0.1	0.356	0.364	0.368	0.373	0.365	0.380	0.378	0.365	0.372
0.05	0.322	0.317	0.317	0.318	0.314	0.311	0.317	0.299	0.301
0.01	0.244	0.250	0.239	0.246	0.226	0.216	0.219	0.198	0.209

Figure 3.5: **Crosslier detection performance across different methods.** Heatmaps depict the AP scores for our method (EXPOSE), LOF, and IF. Performance values were averaged across datasets and random initialisations. In the vertical axis, ρ_y is the proportion of samples which have been category-swapped. In the horizontal axis, ρ_x denotes the proportion of features (in category-swapped samples) of which the values were replaced.

Note that to perform a correct comparison, EXPOSE was not subject to any optimisation: the model selection step was reduced to a single learner with a single set of default hyperparameters. When deployed onto a real-world scenario, model selection should be applied to select the best possible learner and hyperparameter configuration, as described in Section 3.4.

3.7 Discussion

The EXPOSE method is evidently better at detecting crossliers through the exploitation of category models, when compared to standard anomaly detection methods. This was expected, as crossliers are defined based on their feature values in a category-wise manner. High dimensionality and feature dependence are also better dealt with through the appropriate selection of learner with adequate feature selection and regularisation protocols.

The implementation of the EXPOSE method is to be seen as a wrapper over different components: at its core, it is a data-driven category-modelling method using learner functions. Score calibration is applied and, even though a selected model might have a low AUC, the crosslier scores are —we argue— reliable.

For low AUC values, the crosslier scores will tend to cluster at 1 (corresponding to the posterior 0.5). In this sense, EXPOSE will not *expose* a sample unless its respective category is well modelled (high AUC). This property ensures adequate precision of the sample *exposed* and is of particular relevance when dealing with sensitive Inspectorate domains where wrongly-targeting instances has negative outcomes. Assuming sensible feature values and category labels, a high AUC depends only on the learner and hyperparameters selected.

3.8 Chapter Conclusion

In the present work, we (1) defined a specific type of data anomaly, which we term *crosslier*, (2) introduced the EXPOSE method to *crosslier detection*, and (3) designed the *crosslier diagram*, a visualisation tool to represent crossliers evidently. We showed that conventional anomaly detection methods (LOF and IF) are ill-suited for crosslier detection when compared to eXPose.

Although domain-insensitive, EXPOSE produced valuable domain-specific insights into the problem scenario of targeting potentially fraudulent permits of waste transportation across European countries. We defined *crosslier* as an instance which is more similar to other categories than its own; in other words, it is a sample which likely carries company misconduct.

Extensive preprocessing and optimisation steps were performed which culminated in well-performing (high AUC) models of waste categories. Accordingly, the feature values collected in the waste permits allow for suitable differentiation. This finding shows that administrative data allow for compliance checking. After presenting the crosslier diagrams to the inspectors, their assessment was on par with the expected workings of our EXPOSE method: (1) detected crossliers were considered suspicious, and (2) were marked for further inspection. We remark that these cases had gone undetected in standard permit review operations. So, the crosslier diagram was considered by the inspectors a beneficial extension to current methods.

One clear limitation of our experimental setup is, however, that no direct link can be made between (a) hyperparameter optimisation towards AUC performance and (b) crosslier detection performance. While, by definition, it holds that higher classification performance enables more extreme crosslier scores than lower classification performance, the nature of the relationship between the 2 aforementioned points (a) and (b) should be empirically assessed. To this end, the work by [Van Rijn and Hutter, 2018] would prove invaluable towards efficiently selecting the set of hyperparameters over which the optimisation search should be performed.

As a different future research direction, we recommend close cooperation with the inspectors for the following three reasons: (1) by receiving their feedback on the inspected crosslying permits, our method is further validated; (2) we can use the inspected crosslying cases as labelled instances in a supervised learning scenario towards compliance/non-compliance modelling; and (3) EXPOSE is applicable to other problems within the Inspectorate, which further aids the inspectors.

Chapter 4

Noise-Resilient Classifier

Noise in data is a pervasive concern, with causes ranging from human entry errors to flawed automated detection tools. When used to learn a classifier, noisy samples —where class labels and feature values may be corrupted— can seriously deteriorate the resulting classifier performance. In this chapter we propose DENOISE, a unified method to perform classifier learning from noisy samples that leverages noise detection and sample weighting techniques.

The proposed approach consists of learning a noise-resilient classifier through a log-odds sample weighting strategy, in which the weights are derived from the noisy instances in a label noise detection step, as described in Chapter 3. We report on the performance of our method in a controlled scenario where noise was artificially injected into a diverse set of datasets.

Different parameterised configurations of label noise and feature noise proportions were extensively tested against current state-of-the-art methods from the fields of classifier learning under noisy conditions and label noise detection. Results over ten datasets show that overall our method outperforms the state-of-the-art with respect to both learning from noisy data and noise detection, further validating the approach set out in Chapter 3 (which answered RQ2(a)), and most importantly providing an answer to RQ2(b): given data with label noise, how can noisy-samples be used to learn a well-performing model?

The current chapter corresponds to the following publication:
Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2022). Noise-resilient classifier learning. *Pattern Recognition (under review)*

4.1 Noise and Performance Degradation

Noisy data are a prevalent issue in many data-reliant domains. Random input errors, or even malicious intent, cause problematic hurdles for any data descriptive or predictive task. In this chapter, we consider a supervised classification scenario in which we distinguish between feature noise (see Definition 1.16) and label noise (see Definition 1.17).

Samples within a classification dataset may have noise in either (1) the feature values; (2) the class labels; or (3) a combination of both [Wu, 1995]. A major problem of noisy data towards classifier learning is the severe performance degradation of the learned classifier [Heskes, 1994, Wilson and Martinez, 2000, Zhu and Wu, 2004]. A second issue may come from the complexity of the trained model. The impact of the complexity can be assessed by the extent to which a classifier is explainable [Brodley and Friedl, 1999, Segata et al., 2010]. The two problems may present serious consequences such as leading to inaccurate medical diagnoses [Zhang et al., 2006, Holzinger et al., 2017]. targeting the noisy samples is an endeavour in itself. Yet, an adequate detection of such samples can be used to improve classifier learning with noisy data [Gamberger et al., 1999]. In the case where feature noise is assumed, three different approaches have been proposed. First, if feature noise comes in the form of missing values, imputation and missing-indicator methods can be deployed. They allow for adequate learning [Pereira Barata et al., 2019]. Second, if noise relates to the existing values, a viable solution is to apply standard outlier detection methods. As such, samples with outlying values are targeted and consequently discarded prior to learning a classifier [Li et al., 2015]. Third, the usage of learners which are robust to samples with noisy features has been investigated with positive results [Sáez et al., 2014].

Most literature focuses on label noise [Frénay and Verleysen, 2013], as it is typically more detrimental to classifier learning than feature noise. Two reasons are: (1) there is a greater number of features than the single category label, and (2) not all features are equally important towards learning a classifier, whereas the class label of a sample is always of paramount importance [Sáez et al., 2014]. In [Frénay and Verleysen, 2013], the authors categorise label noise into three types or mechanisms:

1. NCAR, the proportion of mislabelled samples is the same per class, and thus independent of features and class;
2. NAR, the proportion of mislabelling is dependent on class, and independent of features;
3. NNAR, the proportion of mislabelled samples may or may not be the same per class and is dependent upon the features.

Joint class and feature noise (i.e., the NNAR case) is often attributed to class overlap. Samples of different classes which are similarly feature-valued increase class-separability issues [Beigman Klebanov and Beigman, 2009]. From a different perspective, a NNAR mechanism might be a result of malicious data manipulation [Pereira Barata et al., 2018b]. As an example, let us consider the practical scenario of international transportation of waste. Transit of wasteful materials is highly regulated through a system of permits. Waste transportation costs vary considerably depending on the waste category of a permit (i.e., class). Therefore, companies have a financial incentive to allege transporting waste with erroneous categorisation (i.e., mislabelling). To further mask the manipulated label, other *permit values* (i.e., features) might be altered to resemble those of the false class label. In other words, the manipulated feature values may contrast the unobserved true label. This complex NNAR scenario is the focus of our research. To address the issue of learning with label noise, at least three strategies have been proposed in the literature. We mention the first two strategies —*baseline* and *preprocessing*— and discuss to some extent the third strategy: *sample weighting*

The first strategy is a *baseline strategy*. It involves using robust noise-tolerant learning architectures [Abellán and Masegosa, 2010] by deploying regularisation and feature selection protocols [Wang et al., 2019, Ghosh et al., 2017]. Its main advantage is the fact that it does not require any data pre-processing. However, using this strategy alone is not sufficient as the information contained in the noisy samples is not considered.

The second strategy is a *preprocessing strategy* in which noisy samples are targeted for either removal from the dataset or for label swapping —data cleaning or cleansing [Miranda et al., 2009]. The action to remove or relabel a sample depends on chosen decision threshold. In practice, this is a difficult choice. It often leads to the case where too many or too few instances are targeted, hence compromising the performance of the classifier [Koplowitz and Brown, 1981].

The third strategy involves using *sample weighting* (i.e., coefficients) in the loss function during learning [Liu and Tao, 2015]. Optimally, sample weights reflect the (un)certainty of the recorded feature values and class labels. However, current weighting approaches present three main hurdles, which we discuss below.

First, a sample weighting strategy may require a separate non-noisy (i.e., curated) sample to compute the weights which is often not available [Ren et al., 2018]. Second, current literature tends to focus on non-negative weights. This may impact the learned model by not taking advantage of the certainty that a sample is probably a mislabel. Third, to assign adequate weights to samples based on their observed feature values and class label, an adequate measure of sample *belongingness* is required which is not easily tractable.

From the literature, belongingness is defined as follows [Kylberg and Sintorn, 2013].

Definition 4.1 – ***Belongingness***

Belongingness is a broad term representing the extent to which a class-labelled sample belongs to the class indicated by its label.

The computation of belongingness is generally addressed by exploiting the aforementioned baseline approach: a robust learner is trained on a dataset —or on a portion of it in a CV manner— and the prediction scores of the learned classifier are used as a measure of sample belongingness [Gamberger et al., 1999, Jeatrakul et al., 2010].

However, overfitting may occur when the entire dataset is used for training which frequently results in poor detection performance. Moreover, even if scores are computed with CV to avoid overfitting, the scores are not comparable across folds since the training sets are not the same.

Our Novel Approach

Given the lacunae in the literature, to improve classifier learning with noisy samples, we propose (1) a novel sample weighting strategy and (2) a new detection method. Since sample weighting is reliant on an appropriate measure of belongingness, a new method to label noise detection is really required. By using of a robust surrogate learner, we propose DENOISE, a sample weighting strategy for learning which leverages the belongingness of samples. As a result, we arrive at the following two contributions.

1. *Learning a classifier* that is resilient to the type of noise in such a way that performance loss is minimal compared to the non-noisy case;
2. *Detecting samples* of which the label is corrupted, in which feature values may be disharmonious with respect to the true (unknown) label.

Obviously, but quite important, the two contributions are related. Classifier learning with noisy data depends on the initial step of detecting samples with label noise.

The structure of the chapter is as follows. Section 4.2 states the terminology and the problem description precisely. Section 4.3 refers to the literature related to our work and reveals the open issues which lead us to our contributions. Section 4.4 describes our methods for classifier learning and the detection of samples with label noise. Section 4.5 describes our experimental setup. Section 4.6 presents the results of our experiments. Finally, Section 4.7 concludes this chapter and provides direction for future research.

4.2 Problem Description

Given a class-labelled dataset, we consider the scenario in which the feature values and target labels have been compromised. Moreover, we focus on the case where feature and class label noise distributions share dependencies; i.e., the NNAR scenario. Under this scenario, a noisy sample is defined as follows.

Definition 4.2 – ***Noisy sample***

A noisy sample under the NNAR scenario is a sample of which the class label and some of its feature values are untrue.

4.2.1 Noise Interpretation

Given a sample with an incorrect class label, we consider inappropriate feature values to be values which explicitly correlate more to the observed incorrect class than to the true unobserved one. The true (unobserved) label is thus further masked. This translates at least to two real-world phenomena: (1) disease-mapping given genetic admixture in populations [Schrider and Kern, 2018, Chen et al., 2014], and (2) data-tampering activities relevant to the risk assessment and fraud-detection domains [Diekmann and Jann, 2010]. Here we remark that this translation is a generalisation of the label noise case as studied in [Müller and Markert, 2019] with the addition of feature noise.

4.2.2 Formal Problem Description

In formal terminology, let $\mathcal{D}$ represent a distribution of a pair of random variables $(X, Y) \in \mathcal{X} \times \{+, -\}$, where $\mathcal{X} \in \mathbb{R}^m$. Let also $(X_1, Y_1), \ldots, (X_n, Y_n)$ be an independent and identically distributed (i.i.d.) sample of $\mathcal{D}$, with $(\tilde{X}_1, \tilde{Y}_1), \ldots, (\tilde{X}_n, \tilde{Y}_n)$ as a sample of the corresponding noisy distribution $\tilde{\mathcal{D}}$.

For a given independently distributed label noise rate $\rho_y = P(Y \neq \tilde{Y})$, we denote the feature noise rate ρ_x as the proportion of m dimensions of which the variable (feature) values are sampled from the distribution conditioned with respect to the noisy label $\tilde{Y}$. Under the described NNAR mechanism, the problem description is:

1. can we predict the label Y for an observation X, given noisy training observations $(\tilde{X}_i, \tilde{Y}_i)$? and
2. can we detect the noisy samples $(\tilde{X}_i, \tilde{Y}_i)$ where $\tilde{Y}_i \neq Y_i$?

4.3 Related Work

There is ample literature in both classifier learning in the presence of noisy samples and label noise detection. Here, we report on relevant work related to supervised learning approaches (Section 4.3.1) and detection techniques (Section 4.3.2) when label noise occurs.

4.3.1 Classifier Learning

Throughout the literature, different approaches have been proposed towards the task of learning with noisy data [Pechenizkiy et al., 2006, Manwani and Sastry, 2013, Teng, 2000, Yin and Dong, 2011, Teng, 2001]. At a broad level, these approaches can be partitioned into three not mutually exclusive categories: (1) robust learners; (2) classification filtering; and (3) sample weighting.

Robust Learners

From the theory [Bartlett et al., 2006] we know that commonly used loss functions in machine learning are not robust to label noise. Yet, some learning architectures and regularisation techniques have been empirically shown to present better results than others in mitigating noisy samples. A breakthrough in this area was [Dietterich, 2000]. There, it was shown that ensemble methods based on bagging achieved superior classification performance when compared to boosting. The reasoning behind is two-fold. First, boosting assigns large scores to mislabelled instances and focuses on those samples to produce the following additive decision boundaries. This leads to poor generalisations. Second, bagging uses different sampling subsets during learning improves on the dissimilarity between the base models making the final classifier more robust.

Still, more recent (gradient) boosting techniques such as LogitBoost and XGBoost have been shown also to be robust to label noise [Gómez-Ríos et al., 2017]. When compared to standard boosting approaches, gradient boosting techniques allow for the misclassification of the training samples rather than over-focusing on them during learning, mitigating overfitting. This factor, in connection with regularisation and feature selection protocols, makes up for greater efficacy when dealing with label noise [Abellán and Moral, 2003]. In summary, it is a challenging research area.

Classification Filtering

Filtering approaches are characterised by either (a) removing or (b) relabelling samples based on a threshold set upon the respective belongingness values.

From empirical experimentation published, the literature has shown that removing samples tends to be more efficient towards learning than relabelling [Cuendet et al., 2007]. However, too many samples might be targeted for removal and that negatively impacts the learning process. Conversely, if too many mislabelled instances are kept, the performance of the learned model is also heavily compromised [Koplowitz and Brown, 1981].

Sample Weighting

A more sophisticated approach to classifier learning with label noise involves using sample weights during learning; this approach is termed *sample weighting*.

Definition 4.3 – ***Sample weighting***

Sample weighting is an approach by which, under a classifier learning scenario, training samples are assigned *weights* according to some specific weighting strategy such that the weights reflect the contribution of each sample towards learning the final classification model.

These weights are applied as coefficients in the loss or error function during risk minimisation. As such, samples have either greater or lesser impact towards learning the final model.

Conceptually, instances with a higher belongingness score (see Section 4.3.2) will have higher weights, and vice-versa. The work presented in [Ren et al., 2018] shows how to estimate sample weights as a minimisation objective by having access to a proportion of non-noisy —curated— samples with ground truth labels. Having access to these data is not always possible, therefore limiting the applicability of this particular sample weighting strategy.

In [Liu and Tao, 2015], the authors demonstrate how any surrogate loss function designed towards a standard classification task can take advantage of sample weighting strategies when noisy labels are present. They propose a weighting strategy based on the ratio of distributions, often used in domain adaptation [Gretton et al., 2009], by assigning a sample weight $\mathcal{W}_i$ to the i^{th} instance based on the following ratio of posterior probabilities:

$$\mathcal{W}_i = \frac{P_{\tilde{\mathcal{D}}}(\tilde{Y}_i|\tilde{X}_i) - \rho_{-y}}{(1 - \rho_+ - \rho_-) \cdot P_{\tilde{\mathcal{D}}}(\tilde{Y}_i|\tilde{X}_i)}, \tag{4.1}$$

$$\rho_{-y} = \min_{\tilde{X} \in \tilde{\mathcal{X}}} P_{\tilde{\mathcal{D}}}(\tilde{Y}|\tilde{X}). \tag{4.2}$$

The drawback of this approach relates to the range of values of the weights. Since all weights are non-negative by definition [Scott, 2015], samples with a high probability of being noisy have a low contribution towards learning.

This characteristic is undesirable since the valuable information contained in those instances is lost. To some extent, it may be detrimental to classification performance, analogous to the removal of samples in the filtering methods.

4.3.2 Label Noise Detection

Most label noise detection methods are based on supervised learning approaches [Frénay and Verleysen, 2013]. Their purpose is two-fold: (1) to target samples which may indicate real-world noncompliance within the inspection domains [Pereira Barata et al., 2018b]; and (2) to perform data preprocessing towards classifier learning [Gamberger et al., 1996].

The task of detecting label-noisy samples is most commonly undertaken by analysing measures of belongingness of a sample towards its observed class label. In this sense, belongingness may be represented as either a score function or the class-conditional posterior probability $P(Y|X)$, both usually tractable by classifier learning [Jeatrakul et al., 2010, Thongkam et al., 2008, Brodley and Friedl, 1996]. Instances with a low score or posterior probability with respect to the class label may be flagged as mislabels [Sun et al., 2007]. Throughout the label noise detection literature, a recurring theme is the usage of supervised classification techniques to infer sample belongingness [Frénay and Verleysen, 2013]. Accordingly, belongingness may be computed according to one of the following three strategies.

The first strategy is to learn a classifier on the entire dataset and to deploy the trained model on the same dataset; in recent work [Müller and Markert, 2019], a robust learner was applied with the purpose of detecting mislabelled entries and presented them to human experts for further evaluation. Even though robust learners may be used, however, overfitting may still occur. Thus, mislabelled instances might be evaluated inappropriately, resulting in unreliable belongingness values.

The second strategy involves learning multiple classifiers on training subsets in a CV manner and deploying each trained model on the corresponding validation set [Gamberger et al., 1999]. While this strategy helps mitigate overfitting, classifier outputs are not comparable across folds since difference training sets were used to yield the scores [Bennett, 2000].

The third strategy is ensemble voting (e.g., majority or consensus by different learners). It can be applied to the generate votes by following either strategy previously mentioned [Miranda et al., 2009,John, 1995]. Since this strategy is dependent on the aforementioned strategies, all mentioned issues apply. A further disadvantage of these approaches is their focus on the removal or relabelling of samples: a poor decision described in Section 4.3.1.

Literature in both classifier learning with noisy samples and label noise detection tasks is extensive. With respect to classifier learning, simply using a robust learner is a too generic approach and does not actively leverage the noisy samples. Removing or relabelling samples is also not suitable since the choice of hard thresholds is difficult to justify.

Sample weighting can mitigate the threshold issue, but current strategies either require access to curated samples, which are hard to obtain, or they do not exploit the information of probable mislabels during training. That is, weight values asymmetrically take belongingness and non-belongingness into account. In terms of label noise detection, there is currently no solution to computing belongingness which provides both minimal overfitting and calibrated (i.e., comparable) output. Ultimately, a new sample weighting strategy is required, as well as a novel sample belongingness computation approach. In the following Section 4.4, we describe our method.

4.4 The DENOISE Method

Below now provide the details of our method: DENOISE. It is data-driven method which provides noise-resilient classification by effectively identifying noisy samples, jointly dealing with the problems of label prediction with noisy samples and label noise detection. Succinctly, it learns a surrogate classifier from noisy data which is robust to both label noise and feature noise collectively by means of a log-odds sample weighting strategy. The sample weights are retrieved when addressing the label noise detection problem.

For each instance we retrieve the calibrated belongingness values and use them to compute the respective sample weights. In turn, the belongingness values are yielded by several robust surrogate learners deployed in a CV fashion with the addition of a calibration protocol. The label noise detection is then solved by applying a sensible threshold to the calibrated output. In Section 4.4.1 we detail our classifier learning setup. In Section 4.4.2 we describe the label noise detection process.

4.4.1 Learning with Sample Weights

The main concept behind our method is directly linked to the strategy by which sample weights are computed. Given noisy samples $(\tilde{X}_1, \tilde{Y}_1) \ldots, (\tilde{X}_n, \tilde{Y}_n)$ of $\tilde{\mathcal{D}}$, a loss $\mathcal{L}$, and weights $\mathcal{W}_1, \ldots \mathcal{W}_n$, the task of the learner is to find a function $f \in \mathcal{F}$:

$$\arg\min_{f \in \mathcal{F}} \frac{1}{n} \cdot \sum_{i=1}^{n} \mathcal{W}_i \cdot \mathcal{L}(f(\tilde{X}_i), \tilde{Y}_i), \text{ where} \tag{4.3}$$

$$\mathcal{W}_i = \ln \left(\frac{P_{\tilde{\mathcal{D}}}(\tilde{Y}_i|\tilde{X}_i)}{1 - P_{\tilde{\mathcal{D}}}(\tilde{Y}_i|\tilde{X}_i)} \right). \tag{4.4}$$

We propose the sample weight of the i^{th} instance to be the log-odds of the event, i.e., the posterior probability; sample weights can therefore take zero, positive, or negative values.

Zero

Sample weights of zero only occur for samples with a posterior probability of 0.5. It entails no contribution to the learning of the final learner. This is sensible, since a posterior of 0.5 towards one class is the same towards the other.

Positive

Weights greater than zero translate to the learner having the information that the observed label is probably not noisy. The larger the weight associated with a specific instance, the more a learner is impacted by it, while *trusting* its label.

Negative

Weights lesser than zero follow the same logic as positive weights except that the observed label is assumed to be incorrect during learning. A negative weight inverts the output of the loss function: the learner is rewarded for incorrectly learning the observed label. The more negative a sample weight is, the larger the impact of that sample towards learning its opposite label.

Posterior Estimation

Albeit intuitive and conceptually simple, our method requires the estimation of posterior probabilities to compute the sample weights; see Eq. 4.4. To be able to compute these posterior probability estimates, and hence the sample weights, we use sample belongingness as a starting point towards acquiring the posteriors. Since belongingness has a myriad of caveats as detailed in Section 4.3.2, in the following Section 4.4.2 we show how to generate it appropriately, addressing the pitfalls mentioned in the literature.

4.4.2 Posterior Estimation and Detection

Our approach to label noise detection is data-driven in the sense that it uses a set of learning functions to compute the belongingness of samples. The belongingness will translate to the posterior probabilities required to compute the sample weights, required for classifier learning (mentioned in Section 4.4.1).

Learning Functions

We follow a supervised learning approach, where the belongingness of a sample is determined by a set of learned classifiers. The learners should be robust to noise and incorporate (1) regularisation, (2) feature selection, or (3) both protocols [Sharma et al., 2017]. The choice of architecture should be sensibly chosen given the type of data being handled; e.g., tabular data may be handled with gradient boosted approaches [Pafka, 2019], and image datasets by dropout convolutional neural network [Park and Kwak, 2016]. To minimise overfitting, belongingness is computed per class on a left out part, rather than on the training set. Consequently, we need to optimise several classifiers in a CV setup.

Since, each CV fold has a specific training set with which a learned classifier is yielded, the output of each classification model is not necessarily comparable. As a result, all classifiers must be calibrated such that their output is comparable. To note, all hyperparameters can be optimised through standard CV [Claesen and De Moor, 2015, Bergstra and Bengio, 2012].

Calibration

To calibrate the output of the learner functions, Platt scaling [Platt et al., 1999] is used. This is a widely accepted method in supervised learning literature which converts classifier output into well-calibrated posterior probabilities [Böken, 2021, Niculescu-Mizil and Caruana, 2005, Guo et al., 2017]. Here, multiple learner functions are learned and calibrated; calibration sets are used such that the output of a learner function is, itself, used as input towards re-learning the observed class label by sigmoid functions (i.e., LR modelling).

The original output of a learned model then becomes the estimated posterior probability through nested CV. For an outer K-fold CV setup, each K-training fold is further split using L-fold CV. The K-training folds serve to calibrate the learner functions. Probabilities are gathered by deploying the calibrated learners onto the respective K-test folds. The estimated posterior probabilities of all samples become the union of all the folds:

$$P_{\tilde{\mathcal{D}}}(\tilde{Y}|\tilde{X}) = \bigcup_{k=1}^{K} P_{\tilde{\mathcal{D}}}(\tilde{Y}_k|\tilde{X}_k). \tag{4.5}$$

$P_{\tilde{\mathcal{D}}}(\tilde{Y}_k|\tilde{X}_k)$ represents the posterior probabilities of samples from the k^{th} test fold, given by the learners calibrated on the respective k^{th} training fold.

Detection

To solve the noisy sample detection problem, we propose using the aforementioned estimated posterior probabilities as detection scores in which the lower the probability of a sample, the more likely that sample is to be flagged as noisy. A detection threshold may be applied to the sample probabilities, such that posterior values lesser than 0.5 flag samples as having label noise. Conversely, a monotonic transformation could be applied such that higher transformed probabilities equate to higher detection scores; e.g., the $-\log_2(x)$ function transforms probabilities into their information content, in which samples with a value higher than 1 are considered label noise.

4.5 Experiments

In this section we describe the experimental setup by which we evaluate our methods. The setup for classifier learning and label noise detection share three similarities:

1. ten noise-free classification datasets were gathered to allow for a controlled scenario, in which ground truth labels are known (Section 4.5.1);
2. to simulate a NNAR mechanism, a proportion of the class labels was flipped and a proportion of feature values were replaced, replicated for several random seeds (Section 4.5.2);
3. baseline state-of-the-art methods were used to gauge the comparative performance of our approaches (Section 4.5.3).

To measure the performance of classifier learning, classifiers were learned on noise-injected training sets, deployed on non-manipulated test sets, and the AUC [Narkhede, 2018] was computed (Section 4.5.3). For label noise detection, the detection targets were considered as the manipulated samples, and AP [Robertson, 2008, Naseer et al., 2018] was used as the performance measure of the detection task (Section 4.5.3).

For the choice of learner, a gradient boosted framework (XGBoost [Chen and Guestrin, 2016]) was selected for its robustness. The surrogate loss function applied was the logistic loss [Painsky and Wornell, 2018]. For reproducibility purposes, our setup is made available online with the all necessary code to download the datasets, manipulate them, perform the learning and detection tasks, and output the yielded results[1].

---

[1] `https://github.com/pereirabarataap/denoise`

4.5.1 Data

Since we are interested in measuring the performance of classifier learning and label noise detection, we gathered datasets with known class labels. Ten benchmark [Asuncion and Newman, 2007] classification datasets were retrieved from *openML* [Vanschoren et al., 2014]: an open, organised, and community-driven online ecosystem for machine learning. These datasets were selected for their heterogeneity regarding sample size, dimensionality (i.e., number of features), and feature type (e.g., number of numerical or categorical features).

Table 4.1 summarises each dataset with respect to aforementioned characteristics. Regarding it, *#samples* indicates sample size, and *#features* represents the number of features of which *#numeric* are numerical and *#category* are categorical. The datasets were then manipulated to simulate a NNAR mechanism.

Table 4.1: Datasets retrieved for noise simulations

ID	#samples	#features	#numeric	#category
1495	250	6	0	6
53	270	13	13	0
40710	303	13	5	8
40690	512	9	0	9
335	554	6	0	6
1510	569	30	30	0
40705	959	44	42	2
1462	1372	4	4	0
1504	1941	33	33	0
41143	2984	144	8	136

4.5.2 Synthetic Noise

Noise was synthetically generated by replacing class labels and feature values. Different combinations of label noise ρ_y and feature manipulation ρ_x were considered. Specifically for the learning task, datasets were first split into training (90%) and testing (10%) sets; only the training sets were injected with noise. Noise was generated ten times with different initialisation seeds per pair (ρ_x, ρ_y) to account for randomness.

Label Noise

Several proportions of label noise were considered. For each dataset, $\rho_y \in \{.05, .1, .15, .2, .25, .3, .35, .4\}$ label noise proportions were introduced following a uniformly-distributed sample selection. This range of values was chosen since $\rho_y < 0.05$ would have negligible impact on robust learners and $\rho_y > 0.4$ would prove too corrupt for any meaningful experimental results.

Feature Manipulation

To recreate the scenario in which feature values are manipulated, samples which were label-swapped had a proportion of their feature values replaced. The proportions $\rho_x \in \{0, .05, .1, .15, .2, .25, .3, .35, .4\}$ of randomly-selected features were selected per sample. Manipulated features had their values replaced per label-swapped sample as described previously. Replacement values were drawn from univariate feature distributions with parameters estimated conditionally from the category being mimicked.

The distributions used to sample the replacement values were modelled as either: (a) the normal distribution $\mathcal{N}(\mu, \sigma)$ for numeric features, with μ and σ as the estimated mean and standard deviation; or (b) the multinomial distribution with estimated event probabilities $\{p_1, p_2, \ldots, p_\pi\}$, π being the number of unique feature values.

4.5.3 Evaluation

We compared DENOISE to current methods of classifier learning with sample weighting and label noise detection. For all tasks, a learning framework was required which was robust to label noise.

Since the datasets in our experiments data are tabular and have heterogeneous characteristics, a gradient boosted framework with a surrogate logistic loss function was selected and applied equally to all scenarios being tested. For the classifier learning task, the sample weighting method in [Liu and Tao, 2015] (LT15) —detailed in Section 4.3.1, Eqs. 4.1 and 4.2— was used as benchmark. For the label noise detection task, the solution presented in [Müller and Markert, 2019] (MM19) —described in Section 4.3.2— was selected as our benchmark.

Learning Performance

Our method DENOISE and LT15 were evaluated using 10-fold crossvalidation. For each fold, the training set (90%) was comprised of noisy samples and the test set (10%) was comprised of non-manipulated samples. To note, only the weighting strategies of each method were evaluated during the learning task.

Both methods had access to the same posterior probability values per sample, which were yielded by DENOISE.

Classification performance was measured as AUC on the non-manipulated tests set for each fold. Accordingly, the true class label is our target prediction. AUC values were averaged across all folds per dataset, across different random seed initialisations, for each noise configuration pair (ρ_x, ρ_y). Each learning strategy being evaluated yielded one mean AUC score per dataset, per noise configuration pair.

Detection Performance

To compare DENOISE to MM19, the detection targets were considered to be the synthetically manipulated instances described previously. The measure of belongingness yielded by each specific method was used as the detection score. Accordingly, detection performance was measured in terms of AP, a common measure used in anomaly detection [Frery et al., 2017].

AP values were averaged across the different random initialisations, for each noise configuration pair (ρ_x, ρ_y). Each detection strategy being evaluated yielded one mean AP score per dataset, per noise configuration pair.

4.6 Results

Here, we present the results of our experiments. We present them in two manners for both tasks, classifier learning (Section 4.6.1), and label noise detection (Section 4.6.2). An *aggregated* presentation —Fig. 4.1 and Fig. 4.2— is comprised of averages across all performance measures yielded for all datasets and noise configurations. We present these results as heatmaps in which each cell corresponds to a noise pair (ρ_x, ρ_y): lighter cell tones indicate higher performance values. Numbers indicate the mean performance score yielded for that specific noise configuration, across all datasets and random initialisations, for each specific task and method used to solve that task.

A performance difference heatmap is also provided, showing comparative changes in performance between the methods being gauged. Similarly to the previous heatmaps, each cell represents the average performance gain of our method for a particular noise configuration, comparatively to the literature baseline usedc. Should we present all datasets and all noise configurations discriminably, a total of 720 performance values would need to be provided which would not be practical. A presentation *per dataset* —Table 4.2 and Table 4.3— relays the performance means and standard deviations yielded by all our experiments for a subset of datasets (IDs 1510, 10705, and 41143) and noise configurations $(0, .05)$, $(0, .4)$, $(.2, .2)$, $(.4, .05)$, and $(.4, .4)$.

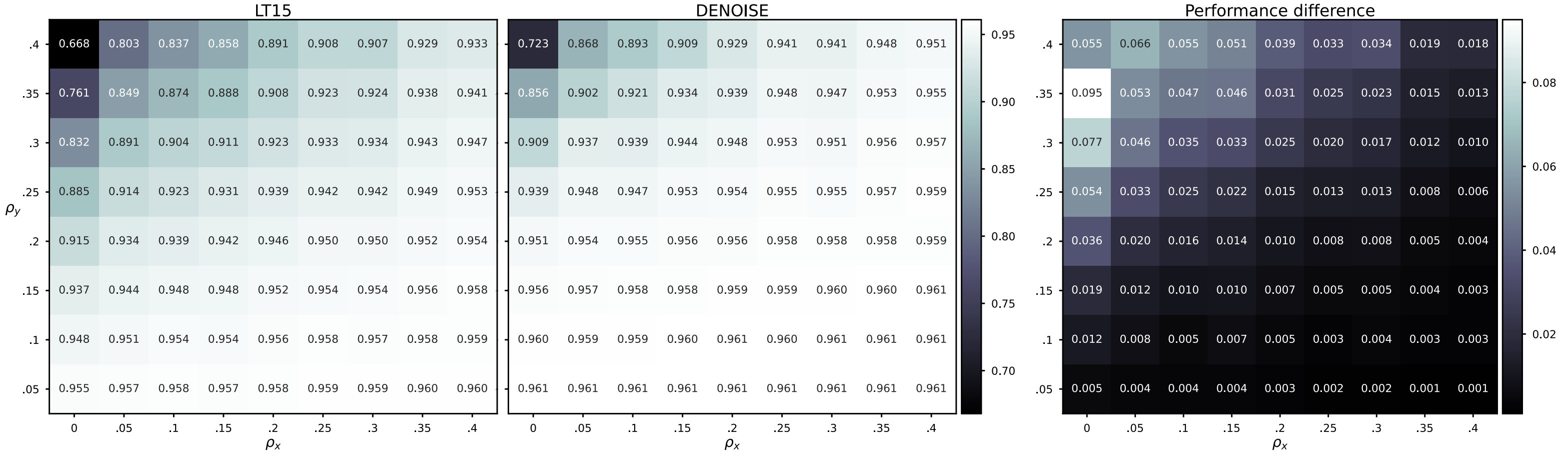


Figure 4.1: **Learning performances with noisy data (aggregated).** The three performance heatmaps represent LT15 (left), our DENOISE method (center), and the performance difference between the two (right). Each heatmap cell depicts the average AUC yielded for the corresponding noise configuration. The vertical axis represents the proportion of samples which have been label-swapped. The horizontal axis denotes the proportion of replaced features values in label-swapped samples.

Table 4.2: Learning performances with noisy data (per dataset)

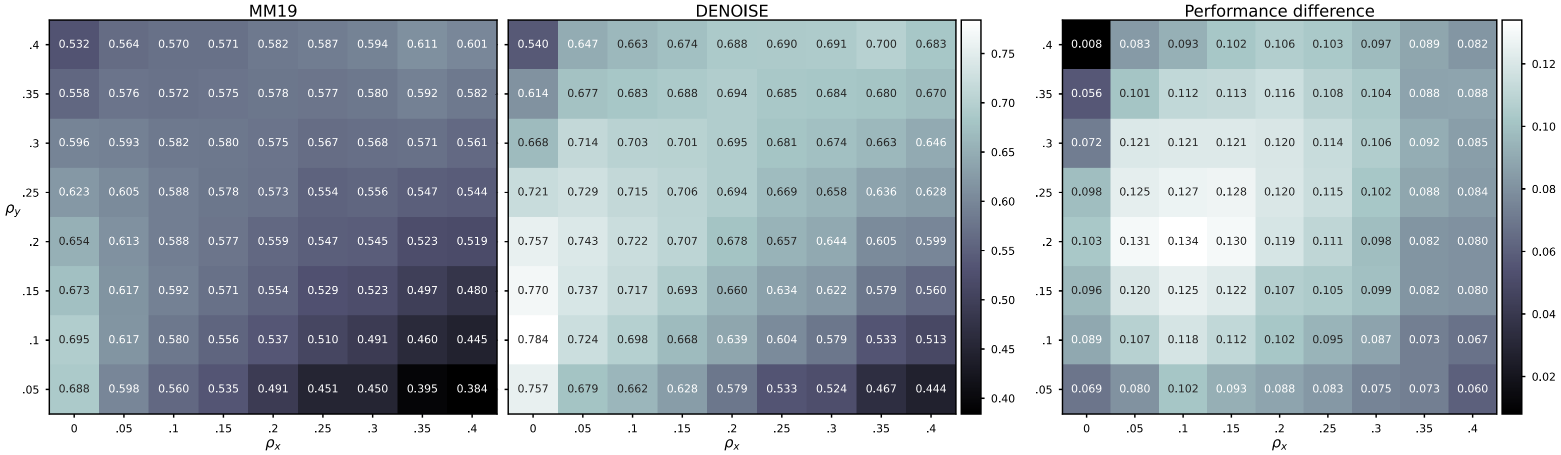
Dataset ID	(ρ_x, ρ_y)	LT15	DENOISE
1510	$(0, .05)$	0.988 ± 0.002	0.987 ± 0.002
	$(0, .4)$	0.727 ± 0.014	**0.860 ± 0.047**
	$(.2, .2)$	0.984 ± 0.003	0.988 ± 0.002
	$(.4, .05)$	0.991 ± 0.001	0.988 ± 0.001
	$(.4, .4)$	0.987 ± 0.003	0.990 ± 0.002
40705	$(0, .05)$	0.968 ± 0.003	0.976 ± 0.002
	$(0, .4)$	0.710 ± 0.016	**0.827 ± 0.072**
	$(.2, .2)$	0.966 ± 0.003	0.967 ± 0.004
	$(.4, .05)$	0.978 ± 0.001	0.976 ± 0.001
	$(.4, .4)$	0.969 ± 0.004	0.962 ± 0.008
41143	$(0, .05)$	0.858 ± 0.003	0.865 ± 0.002
	$(0, .4)$	0.653 ± 0.012	**0.753 ± 0.014**
	$(.2, .2)$	0.854 ± 0.002	0.857 ± 0.002
	$(.4, .05)$	0.864 ± 0.002	0.867 ± 0.002
	$(.4, .4)$	0.847 ± 0.005	0.849 ± 0.006

These datasets and noise configurations were selected to provide a representative sample of the entire set of experiments. The datasets have mostly disparate characteristics (see Table 4.1), and the noise configurations values are the most spread. Bolded values indicate a mean performance gain of at least 0.01 of the corresponding method versus the other.

4.6.1 Classifier Learning Task

We present the *aggregated* average AUC performance scores yielded for the learning task in Fig. 4.1. The three heatmaps represent LT15 (left), DENOISE (center), and the performance difference between the two methods (right). For every configuration of label swapping (vertical axis) and feature manipulation (horizontal axis), DENOISE shows superior performance. The difference in AUC performance varies significantly across different noise configurations (ρ_x, ρ_y).

The minimum performance change is $\approx .01$, seen in, for example, cell $(.4, .05)$. The maximum performance difference is $\approx .1$ in configuration $(0, .35)$. On average, the difference in performance tends to increase as the noise label proportion increases; i.e., the more label noise is present, the better our method fares. The performance difference also tends to increase as the feature manipulation proportion decreases.



MM19

ρ_y \ ρ_x	0	.05	.1	.15	.2	.25	.3	.35	.4
.4	0.532	0.564	0.570	0.571	0.582	0.587	0.594	0.611	0.601
.35	0.558	0.576	0.572	0.575	0.578	0.577	0.580	0.592	0.582
.3	0.596	0.593	0.582	0.580	0.575	0.567	0.568	0.571	0.561
.25	0.623	0.605	0.588	0.578	0.573	0.554	0.556	0.547	0.544
.2	0.654	0.613	0.588	0.577	0.559	0.547	0.545	0.523	0.519
.15	0.673	0.617	0.592	0.571	0.554	0.529	0.523	0.497	0.480
.1	0.695	0.617	0.580	0.556	0.537	0.510	0.491	0.460	0.445
.05	0.688	0.598	0.560	0.535	0.491	0.451	0.450	0.395	0.384

DENOISE

ρ_y \ ρ_x	0	.05	.1	.15	.2	.25	.3	.35	.4
.4	0.540	0.647	0.663	0.674	0.688	0.690	0.691	0.700	0.683
.35	0.614	0.677	0.683	0.688	0.694	0.685	0.684	0.680	0.670
.3	0.668	0.714	0.703	0.701	0.695	0.681	0.674	0.663	0.646
.25	0.721	0.729	0.715	0.706	0.694	0.669	0.658	0.636	0.628
.2	0.757	0.743	0.722	0.707	0.678	0.657	0.644	0.605	0.599
.15	0.770	0.737	0.717	0.693	0.660	0.634	0.622	0.579	0.560
.1	0.784	0.724	0.698	0.668	0.639	0.604	0.579	0.533	0.513
.05	0.757	0.679	0.662	0.628	0.579	0.533	0.524	0.467	0.444

Performance difference

ρ_y \ ρ_x	0	.05	.1	.15	.2	.25	.3	.35	.4
.4	0.008	0.083	0.093	0.102	0.106	0.103	0.097	0.089	0.082
.35	0.056	0.101	0.112	0.113	0.116	0.108	0.104	0.088	0.088
.3	0.072	0.121	0.121	0.121	0.120	0.114	0.106	0.092	0.085
.25	0.098	0.125	0.127	0.128	0.120	0.115	0.102	0.088	0.084
.2	0.103	0.131	0.134	0.130	0.119	0.111	0.098	0.082	0.080
.15	0.096	0.120	0.125	0.122	0.107	0.105	0.099	0.082	0.080
.1	0.089	0.107	0.118	0.112	0.102	0.095	0.087	0.073	0.067
.05	0.069	0.080	0.102	0.093	0.088	0.083	0.075	0.073	0.060

Figure 4.2: **Detection performances of noisy samples (aggregated).** The three performance heatmaps represent MM19 (left), our DENOISE method (center), and the performance difference between the two (right). Each heatmap cell depicts the average AP yielded for the corresponding noise configuration. The vertical axis represents the proportion of samples which have been label-swapped. The horizontal axis denotes the proportion of replaced features values in label-swapped samples.

Table 4.3: Detection performances of noisy samples (per dataset)

Dataset ID	(ρ_x, ρ_y)	MM19	DENOISE
1510	$(0, .05)$	0.858 ± 0.037	0.860 ± 0.037
	$(0, .4)$	0.576 ± 0.014	$\mathbf{0.621 \pm 0.034}$
	$(.2, .2)$	0.719 ± 0.030	$\mathbf{0.837 \pm 0.020}$
	$(.4, .05)$	0.476 ± 0.072	$\mathbf{0.536 \pm 0.052}$
	$(.4, .4)$	0.763 ± 0.018	$\mathbf{0.825 \pm 0.018}$
40705	$(0, .05)$	$\mathbf{0.734 \pm 0.033}$	0.721 ± 0.032
	$(0, .4)$	0.569 ± 0.018	$\mathbf{0.627 \pm 0.042}$
	$(.2, .2)$	0.458 ± 0.029	$\mathbf{0.616 \pm 0.025}$
	$(.4, .05)$	0.196 ± 0.009	$\mathbf{0.322 \pm 0.020}$
	$(.4, .4)$	0.494 ± 0.016	$\mathbf{0.609 \pm 0.022}$
41143	$(0, .05)$	0.224 ± 0.033	$\mathbf{0.464 \pm 0.019}$
	$(0, .4)$	0.491 ± 0.011	$\mathbf{0.533 \pm 0.027}$
	$(.2, .2)$	0.260 ± 0.006	$\mathbf{0.436 \pm 0.011}$
	$(.4, .05)$	0.062 ± 0.003	$\mathbf{0.180 \pm 0.014}$
	$(.4, .4)$	0.364 ± 0.007	$\mathbf{0.503 \pm 0.010}$

In Table 4.2 we show the *per dataset* average AUC performance scores for the selected datasets and noise configurations, as well as the respective standard deviations along the ten different random initialisations. Bolded entries translate to an increase in mean performance of at least 0.01. The differences in performance are most apparent in noise configuration $(0, .4)$ in all datasets, where our method DENOISE vastly outperforms LT15. In those cases, the performance gain is consistently ≥ 0.1.

Regarding the standard deviations, most noise pair configurations present similar values. However, two outliers stand out in the DENOISE entries. Noticeably, datasets with IDs 1510 and 40705, since the same noise configuration $(0, .4)$ shows larger standard deviations than the LT15 method. However, it is also for those two datasets and the particular noise configuration that LT15 also has an increased spread of the mean performance comparatively to all other datasets and noise pairs.

4.6.2 Label Noise Detection Task

We display the *aggregated* average AP performance yielded for the detection task in Fig. 4.2. The three heatmaps represent MM19 (left), our DENOISE method (center), and the performance difference between the two methods (right).

For every noise configuration, our method shows superior performance overall. The difference in AP performance varies significantly across different noise configurations (ρ_x, ρ_y). The minimum performance change is $\approx .01$, seen in, for example, cell $(0, .4)$. The maximum performance difference is $\approx .13$ in configuration $(.1, .2)$. Comparatively, the performance difference steadily decreases from its maximum, regardless of the direction taken in the horizontal or vertical axes.

Table 4.3 represents the *per dataset* AP performance scores for the selection of datasets and noise configurations and standard deviations across the ten random initialisations. With the exception of two entries, our approach yielded a mean performance gain overall. The differences in performance vary across datasets as well as along the noise pairs. For the majority of entries, our method shows higher standard deviation values. Yet, the differences are small. The largest difference in spread between the two methods is 0.024 for the entry with ID 40705 and noise pair $(0, .4)$.

4.7 Chapter Conclusion

In this chapter we proposed DENOISE, a method for noise-resilient classifier learning which leverages label noise detection via log-odds sample weighting. We compared our method to the state-of-the-art in learning with noise and label noise detection, under a NNAR mechanism in which label noise and feature noise may share dependencies.

In summary, with regard to the problem description, we may conclude that in the NNAR scenario DENOISE achieves overall (1) better class label predictions with noisy training data, and (2) better detections of those noisy samples than current literature methods.

We designed an experimental setup in which ten datasets with heterogeneous characteristics were used, representing different domains. For each dataset, different parameterised combinations of label noise and feature noise were extensively explored. Experimental setups were repeated ten times with different random initialisations.

Within the NNAR setting we considered the two tasks of: (1) learning a classifier that is resilient to this type of noise such that performance loss is minimal compared to the non-noisy case, and (2) detecting samples of which the label is corrupted, while the feature values may be disharmonious with respect to the true (unobserved) label. The results demonstrate that DENOISE overall outperforms the state-of-the-art overall in both learning and detection tasks.

Synthesising the NNAR mechanism is a complex task. On the one hand, corrupting class labels in a binary classification setting is rather straightforward.

On the other hand, the addition of feature manipulation involves applying dependency assumptions. While we chose a random feature selection strategy, a different approach could have been chosen; e.g., features could have been selected per class instead of per sample. A different approach to feature selection would be to follow a non-random selection strategy based on feature importance. Similarly, different robust classifiers could have been used, alongside different types of data, as well we different loss functions instead of the applied logistic loss. As such, our results are bound to our experimental setup and further work may be performed to other experimental setups. Yet, we have laid out a framework upon which experimental design choices can be made to generate specific noise scenarios.

Ultimately, handling noisy data remains a difficult task, even though noise mechanisms can be formally defined. For a NNAR case, as discussed in this chapter, the properties of the underlying distributions of the observed noisy data are often varied and not fully tractable. Future work could improve upon these simulations by using different assumptions over the noise in the data, e.g., by stipulating (1) different types of distributions for feature manipulation, and (2) varying correlations with the well-defined class labels.

Chapter 5

Fair Tree Classifier

When learning classification models from biased data, the resulting classifiers tend to exacerbate the biases present [Richardson, 2022]. With respect to the Inspectorate, a case in point is *confirmation* bias.

Consider the following example. In international cargo ship risk assessment, a prevailing trait towards selecting a ship for inspection is the colour of the flag of the ship. A reputable country is assigned a white flag. However, the flag may be either white, grey, or black, and reflects the detention rate of ships for that country. Indeed, inspectors may be disproportionately influenced by the colour of a flag, causing more frequent and stringent inspections of ships with non-white flags, leading to confirmation bias in data.

To learn a classifier from such biased data, the standard classification problem becomes three-fold: (1) it is necessary to learn a model with high classification performance; (2) the impact of the biases on the model must be suppressed (i.e., model fairness); and (3) the performance-fairness trade-off must be tunable such that the requirements by the relevant stakeholders can be easily met.

In this chapter, we propose SCAFF: a solution to the problem at hand in the form of a compound splitting criterion which combines (a) AUC, (b) strong demographic parity, and (c) a performance-fairness trade-off tunability parameter. In our experimental results, we show via performance-fairness trade-off curves how SCAFF generates effective models with competitive performance and high fairness. This result answers RQ3: how can we, from biased data, learn a model tunable with respect to the performance-fairness trade-off such that the selection of the trade-off point is made intuitive for the relevant stakeholders?

The current chapter corresponds to the following publication:
Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2022). Fair tree classifier using strong demographic parity. *Machine Learning (under review)*

5.1 Algorithmic Fairness

The application of machine learning algorithms for classification has become ubiquitous within an abundance of domains [Brink et al., 2016, Sarker, 2021, Azar and El-Metwally, 2013, Pereira Barata et al., 2021, Dressel and Farid, 2018]. Great dependency on automated decision-making, however, gives rise to concerns over model bias; e.g., bias was reported by Amazon's automatic recruitment tool in which women unfairly scored lower. It turns out that models were trained on resumes submitted mostly by men, thus disadvantaging women a priori [Dastian, 2018]. To prevent the modelling of historical biases, it is of the utmost importance to develop fairness-aware methods [European Commission, 2019c].

A fair classification model has three goals: (1) to make adequate class predictions from *unseen* observations; (2) to ensure that the bias in data is suppressed from those predictions [Cho et al., 2020]; and (3) to allow for the tunability of the inherent trade-off between the aforementioned two goals —the performance-fairness trade-off [Kleinberg et al., 2016]— such that the ethical, legal, and societal needs of the end user (i.e., domain expert) are met. Here we remark that the third goal is of greatest importance, as achieving it provides a manner by which trade-off points can be made selectable by the relevant stakeholders.

To quantify model fairness (i.e., the extent to which the biases in data have been suppressed) different fairness measures have been proposed (see Definitions 1.5, 1.6, and 1.7). Traditionally, fairness measures such as demographic parity [Dwork et al., 2012], equal opportunity [Corbett-Davies and Goel, 2018], or equalised odds [Hardt et al., 2016] are used. These fairness measures are all *threshold-dependent*. A threshold-dependent fairness measure is defined as follows.

Definition 5.1 – ***Threshold-dependent fairness measure***

A threshold-dependent fairness measure is a quantification of algorithmic bias with respect to some sensitive group, measured as a function of the class predictions induced by applying a threshold to the (continuous) model output.

Considering a classification model with continuous output, a decision threshold must be set to produce class predictions, upon which those measures are reliant. In other words, fairness would only be ensured with respect to that particular threshold. To counter this limitation, a *threshold-independent* fairness measure can be used instead. A threshold-independent fairness measure is defined as follows.

Definition 5.2 – ***Threshold-independent fairness measure***

A threshold-independent fairness measure is a quantification of algorithmic bias with respect to some sensitive group, measured as a function of the (continuous) model output, rather than the class predictions.

One such measure is the *strong demographic parity*. The strong demographic parity extends the aforementioned demographic parity by considering fairness throughout the entire range of possible decision thresholds. Although having been proposed in [Jiang et al., 2020], the authors provided an implementation of strong demographic parity merely towards the linear classifier case.

Tree-based algorithms are regarded as a state-of-the-art solution for the classification problem [Zabihi et al., 2017, Dogru and Subasi, 2018, Angenent et al., 2020]. Their prevalence in the literature is mostly due to (1) model interpretability, (2) their tendency to not overfit when used as ensembles, (3) requiring little data pre-processing, and (4) handling mixed data types and missingness [Dogru and Subasi, 2018]. Past work on tree splitting criteria has shown positive results with respect to threshold-dependent fairness [Kamiran et al., 2010]. There is a desire to extend it towards the threshold-independent case.

In this work, we propose SCAFF: the Splitting Criterion AUC For Fairness. SCAFF allows for fair tree classifier learning by directly optimising for the threshold-independent fairness measure of strong demographic parity. In particular, we propose a *fair tree classifier* learning algorithm which simultaneously (1) optimises for threshold-independent classification performance (i.e., AUC); (2) suppresses the impact of bias directly in terms of strong demographic parity; and (3) is tunable with respect to the performance-fairness trade-off during learning. In addition, our method handles various multicategorical sensitive attributes simultaneously, and easily extends to bagging (i.e., random forest) and (gradient) boosting frameworks.

The structure of the chapter follows: Section 5.2 expresses our problem description formally; Section 5.3 discusses related work; Section 5.4 elaborates our SCAFF method; Section 5.5 describes our experiments; Section 5.6 refers to our results; and Section 5.7 concludes and recommends research directions.

5.2 Problem Description

We consider the scenario in which a labelled dataset is intrinsically biased with respect to one or more sensitive attributes of which the values may be either binary or multicategorical. Our task is to learn a fair predictive model from the biased data, such that future predictions are independent from the sensitive attribute(s).

We require that the measures of model performance and fairness do not depend on a decision threshold set upon the output. Since there is no unique solution in the trade-off between performance and fairness, the fair model must also be readily tunable in this regard, as to meet the requirements of the application domain.

Formally, consider a dataset D with n samples, m features, and two classes. Without loss of generality, assume the case in which a single binary sensitive attribute exists. Let X, Y, and S be the underlying variable distributions representing the feature space, classes, and sensitive attribute, respectively, from which the n samples were drawn. Accordingly, each sample may be represented as (x_i, y_i, s_i), for $i = 1, 2, \dots, n$.

The goal of the learning algorithm is to learn the distribution for which the conditional $P(Y|X) \approx P(Y|X, S)$. In practice, this amounts to learning from the data a mapping function $f : x \in X \rightarrow z \in Z$ where Z represents the model output (i.e., classification score) upon which a threshold t induces a class prediction, and under which the condition of strong demographic parity must be met, $\forall t \in Z : P(Z \geq t|S_+) = P(Z \geq t|S_-)$, while maximising for the threshold-independent classification performance $P[(Z|Y_+) \geq (Z|Y_-)]$. The compromise between strong demographic parity and the corresponding maximal predictive performance must also be tunable.

5.3 Related Work

In this section, we discuss the concepts from the literature related to our work: the measures of fairness (Section 5.3.1), and the fair tree splitting criteria used towards fair tree classification learning (Section 5.3.2).

5.3.1 Measures of Fairness

Fairness measures in the literature may be categorised as being either (a) threshold-dependent or (b) threshold-independent. With respect to threshold-dependent measures, the three most prevalent are: (1) demographic parity [Dwork et al., 2012]; (2) equal opportunity [Corbett-Davies and Goel, 2018]; and (3) equalised odds [Hardt et al., 2016].

First, *demographic parity* (see Definition 1.6) is the condition under which each sensitive group (e.g. male/female) should be granted a positive outcome, at equal rates. It is the absolute difference between the proportion of positive class predictions $\hat{Y}_+$ in samples with a positive sensitive attribute value S_+ and samples with a negative sensitive attribute value S_-, and is computed as $|P(\hat{Y}_+|S_+) - P(\hat{Y}_+|S_-)|$.

Second, the measure of *equal opportunity* is defined as follows.

Definition 5.3 – ***Equal opportunity***

> Equal opportunity is the fairness measure which considers the absolute difference between the conditional TPR of each sensitive group

Equal opportunity is the fairness measure which accounts for the predictive reliability within each sensitive group and is computed as the absolute difference $|P(\hat{Y}_{+}|S_{+},Y_{+}) - P(\hat{Y}_{+}|S_{-},Y_{+})|$.

Third, the definition of *equalised odds* follows.

Definition 5.4 – ***Equalised odds***

> Equalised odds is the fairness measure which considers the absolute difference between the conditional TPR of each sensitive group, as well as the difference between the conditional FPR of each sensitive group.

Equalised odds extends from the measure of equal opportunity by also incorporating the unreliability of predictions in the sensitive groups. It is computed as $||P(\hat{Y}_{+}|S_{+},Y_{+}) - P(\hat{Y}_{+}|S_{-},Y_{+})| - |P(\hat{Y}_{+}|S_{+},Y_{-}) - P(\hat{Y}_{+}|S_{-},Y_{-})||$.

Albeit computationally different, the three measures share at least one common aspect: the output of the classification model must be binary; i.e., a decision threshold must be placed upon the continuous output which induces the class prediction. As a result, a problem arises when applying these measures towards learning a fair classifier. These measures of fairness are limited to being exclusively reliable for the specific threshold which produces the class prediction: there is no guarantee that fairness holds for different threshold values.

In practice, when learning several fair classifiers for real-world applications, (i.e., hyperparameter optimisation), the final classification model should not be dependent on any arbitrary threshold, as fairness should be maintained throughout. Rather, the decision threshold should only be placed a posteriori, according to the performance requirements of the end user (e.g., precision vs recall) whilst incurring minimal impact over fairness.

With respect to threshold-independent fairness measures, the notion of demographic parity has been extended into strong demographic parity (see Definition 1.7). Strong demographic parity takes into account the continuous output of the model, such that the ordering of the output should be independent of the sensitive groups. It is computed as the absolute difference between the probabilities $|P[(Z|S_{+}) \geq (Z|S_{-})] - P[(Z|S_{+}) < (Z|S_{-})]|$.

Although strong demographic parity was proposed with a working fair learning framework in [Jiang et al., 2020], their implementation only considers the linear classifier case. We focus on extending the implementation towards non-linear models, specifically towards tree-based architectures.

5.3.2 Fair Tree Splitting Criteria

The practice of learning a tree classifier from biased data is directly linked to the splitting criterion used to construct the tree structure. Within the fairness literature with respect to tree-based algorithms, we recommend the works by [Kamiran et al., 2010] and [Zhang and Ntoutsi, 2019], in which different approaches are used to measure classification performance and fairness. The measures are then jointly used as splitting criteria during training to select the best split.

In the work by [Kamiran et al., 2010], the authors propose to address the fair splitting criterion problem, by accounting for the impact of bias in the model during learning. They do so by extending the concept of information gain in traditional classification towards the sensitive attribute. Given data D, a split is evaluated as the information gain with respect to the class label:

$$IG_Y = H_Y(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} \cdot H_Y(D_i), \tag{5.1}$$

and the information gain with respect to the sensitive attribute:

$$IG_S = H_S(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} \cdot H_S(D_i), \tag{5.2}$$

where H_Y and H_S denote the entropy with respect to the class label and the sensitive attribute, respectively, and $D_i, i = 1, \dots, k$ denotes the partitions of D induced by the split under evaluation.

Both information gains are then merged to produce two distinct compound splitting criteria by either: (1) subtracting IG_Y by IG_S, hereinafter termed $\text{Kamiran}_{\text{Sub}}$, or (2) dividing IG_Y by IG_S, hereinafter denoted as $\text{Kamiran}_{\text{Div}}$. Although this work was fundamental in establishing fair tree-learning frameworks, it is limited in scope since fairness is only considered as the threshold-dependent demographic parity.

In the work of [Zhang and Ntoutsi, 2019], a fairness-aware Hoeffding tree (FAHT) is introduced. Although the method was developed with online streaming classification as its focus, the splitting criterion developed may be generally applicable to the fair learning problem. The FAHT approach relies, as with the previous work, on a compound criterion composed of a class label part and a sensitive attribute part and addresses demographic parity. Both works use the same class label information gain IG_Y. However, the fairness component is computed differently between them. For FAHT, the fairness gain FG of a split is given as a function of $Disc(D)$ of a set of data:

$$FG = Disc(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} \cdot Disc(D_i). \tag{5.3}$$

The bias term is defined as the observed demographic parity of the system $|P(Y_+|S_+) - P(Y_+|S_-)|$. The splitting criterion of FAHT evaluates as follows:

$$\begin{cases} IG_Y, & \text{if } FG = 0 \\ IG_Y \cdot FG, & \text{otherwise} \end{cases}. \tag{5.4}$$

The two proposed splitting criteria present some limitations, three of which deserve to be named in particular: (1) the construction processes were developed with only threshold-dependent fairness in mind; (2) both implementations only address a single binary sensitive attribute; and (3) there exists no performance-fairness trade-off tuning parameter built into the splitting criteria. In the following section, we propose our method which lifts these limitations.

5.4 The SCAFF Method

In this section we propose our SCAFF method. It is a probabilistic learning framework which (1) optimises for threshold-independent classification performance (i.e., AUC); (2) addresses fairness in terms of strong demographic parity; and (3) is tunable with respect to the performance-fairness trade-off. In addition, SCAFF leverages multiple sensitive attributes simultaneously and easily extends to bagging and boosting frameworks.

We begin by addressing the implementation of the classification performance in Section 5.4.1, followed by the implementation of the fairness measure of strong demographic parity in Section 5.4.2. In Section 5.4.3, we provide our compound splitting criterion which incorporates a tunable parameter towards the trade-off between classification performance and fairness. In Section 5.4.4, we describe the tree construction process, reporting on how our method leverages multiple sensitive attributes simultaneously and extends to bagging and boosting frameworks. A working Python implementation of our algorithm can be found in [Pereira Barata, 2021].

5.4.1 AUC Computation

In machine learning, the AUC is a measure which expresses the quality of a sample ordering with respect to a binary label $\{Y_-, Y_+\}$. It computes the probability $P[(Z|Y_+) \geq (Z|Y_-)]$. Here, a random order results in AUC $= 0.5$ and a perfect order results in AUC $= 1$; conversely AUC $= 0$ if all labels are flipped and still perfectly ordered.

Traditionally, computing the AUC has a time complexity $O(n \cdot \log(n))$:

$$\mathrm{AUC}(Z,Y) = \frac{\sum_{i=1}^{y_+} \sum_{j=1}^{y_-} \sigma(Z_i, Z_j)}{y_+ \cdot y_-}, \tag{5.5}$$

where

$$\sigma(Z_i, Z_j) = \begin{cases} 1, & \text{if } Z_i > Z_j \\ \frac{1}{2}, & \text{if } Z_i = Z_j \\ 0, & \text{otherwise} \end{cases}. \tag{5.6}$$

Here, y_+ and y_- are the number of all instances Y_+ and Y_- respectively, and Z_i and Z_j represent the Z output scores of each corresponding instance.

Yet, for the scenario in which a parent node is split into two child nodes —towards candidate split evaluation—the time complexity of computing the AUC may be reduced. From [Lee, 2019], the AUC of a split may be re-written as a function of the TPR and the FPR induced by the split. The AUC then becomes:

$$\mathrm{AUC} = \frac{1 + \mathrm{TPR} - \mathrm{FPR}}{2}. \tag{5.7}$$

For each candidate split, the child node with highest $P(Y_+)$ is assigned as the positive prediction node such that all samples contained in it are labelled $\hat{Y}_+$. The other child node induces $\hat{Y}_-$. This strategy is equivalent to computing the AUC traditionally; i.e., assigning samples in each node with Z scores equal to the proportion of ground truth positive labels $P(Y_+)$ of their corresponding node. Hereinafter, we denote $\mathrm{AUC_Y}$ as the AUC with respect to the class label.

5.4.2 Strong Demographic Parity

The strong demographic parity condition aims to minimise the difference in candidates from the sensitive groups among the selected candidates, regardless of any arbitrary decision threshold t. The goal is to minimise the expression $|P[(Z|S_+) \geq (Z|S_-)] - P[(Z|S_+) < (Z|S_-)]|$ from Section 5.3.1. The condition is reached by learning the target function f which randomly orders the samples towards the sensitive groups; i.e., the AUC towards the sensitive attribute.

We find the fair classifier f by optimising for an AUC value of 0.5 on the sensitive attribute. In order to solve the optimisation problem, we aim at minimising the AUC with S_+ as the positive class, which we denote as $\mathrm{AUC_{S_+}}$.

Since $\mathrm{AUC}_{\mathrm{S}_+} = 0$ is as maximally unfair as $\mathrm{AUC}_{\mathrm{S}_-} = 1$, we define *sensitive AUC* ($\mathrm{AUC}_{\mathrm{S}}$) —$f_S$ from Section 5.2— as the following:

$$\mathrm{AUC}_{\mathrm{S}} = \max[1 - \mathrm{AUC}(Z, S), \mathrm{AUC}(Z, S)], \tag{5.8}$$

such that the $\max$ operator bounds the range of possible AUC_S values to $[0.5, 1]$.

Definition 5.5 – ***Sensitive AUC***

Sensitive AUC is the AUC towards a sensitive attribute, bounded to values $[0.5, 1]$ and is proportional to the strong demographic parity. $\mathrm{AUC}_{\mathrm{S}}$ can be computed as a function of the strong demographic parity:

$$\mathrm{AUC}_{\mathrm{S}} = \frac{\text{strong demographic parity} + 1}{2}$$

$\mathrm{AUC}_{\mathrm{S}}$ of 1 indicates that the model is completely biased, while 0.5 indicates that the model is complete fair.

Now that both classification performance $\mathrm{AUC}_{\mathrm{Y}}$ and fairness measure $\mathrm{AUC}_{\mathrm{S}}$ have been described, the splitting criterion may be constructed.

5.4.3 Splitting Criterion AUC For Fairness

Towards tunability of the performance-fairness trade-off, we define the *orthogonality* parameter Θ as follows.

Definition 5.6 – ***Orthogonality parameter*** Θ

The orthogonality parameter $\Theta \in [0, 1]$ is the parameter of SCAFF which regulates the performance-fairness trade-off of the learned model: $\Theta = 0$ results in a completely biased but most performing model, whereas $\Theta = 1$ results in a completely fair but nonperforming model.

The objective is then to find a split which, for a given Θ, maximises $\mathrm{AUC}_{\mathrm{Y}}$ (towards $\mathrm{AUC}_{\mathrm{Y}} = 1$), while minimising $\mathrm{AUC}_{\mathrm{S}}$ (towards $\mathrm{AUC}_{\mathrm{S}} = 0.5$). Accordingly, for the fair classification problem given instance scores Z, class label Y, and sensitive attribute S, we define SCAFF:

$$\mathrm{SCAFF}(Z, Y, S, \Theta) = (1 - \Theta) \cdot \mathrm{AUC}_{\mathrm{Y}} - \Theta \cdot \mathrm{AUC}_{\mathrm{S}}. \tag{5.9}$$

The purpose of Θ is to change the direction of the splitting criterion score towards either classification or fairness. To illustrate this effect, consider Fig. 5.1.

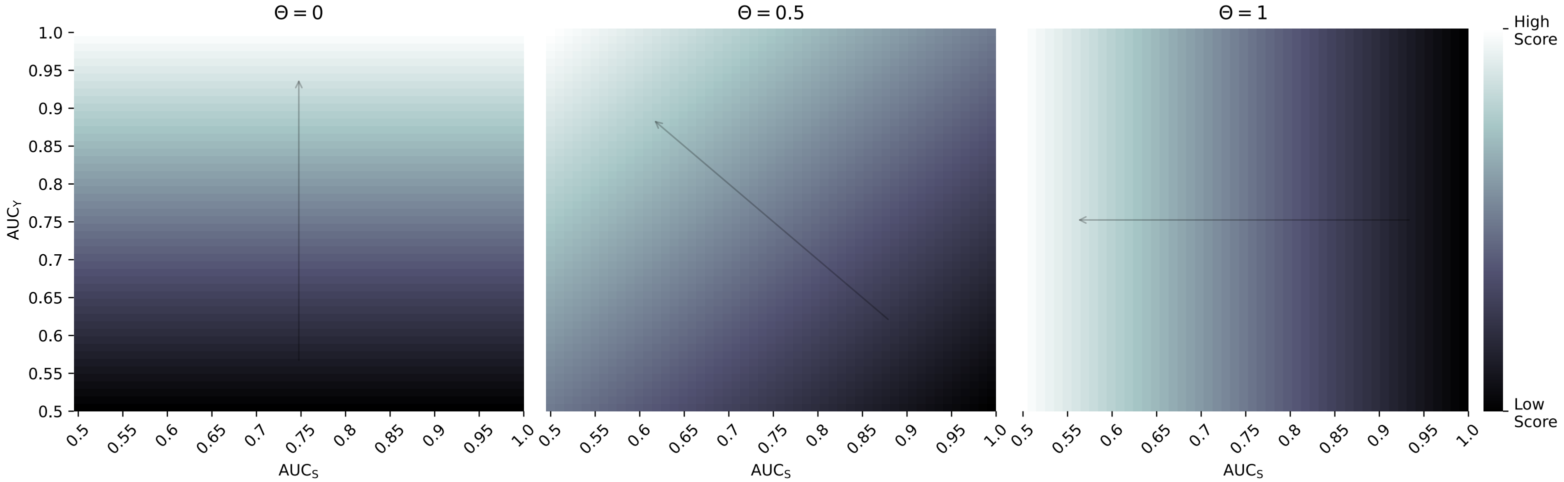


Figure 5.1: **Effect of the orthogonality Θ over SCAFF scores.** Each heatmap depicts the SCAFF scores at three different values of Θ. Each heatmap cell corresponds to a specific combination of AUC_Y (vertical axis) and AUC_S (horizontal axis) from which the score is computed.

Each heatmap represents, for varying values of Θ, the split evaluation scores for all possible values of AUC_Y (vertical axis) and AUC_S (horizontal axis), according to Eq. 5.9. The direction of the optimal score, from darkest to brightest tones, is additionally represented as an arrow. From left to right, the optimal score direction rotates along Θ. We call it the orthogonality parameter since it rotates the direction of the optimal scores, making $\Theta = 0$ and $\Theta = 1$ orthogonal score directions.

5.4.4 Tree Construction

As with any typical tree architecture, learning is done by selecting, at each step (i.e., depth), the split which optimises the splitting criterion score. A split at some feature value partitions a node into two child nodes and is evaluated according to the Z scores of the parent node and the new Z' scores of the child nodes induced by that split. The optimal split is the one which, across all possible feature value split points, maximises the splitting criterion score.

Given (a) the parent node scores Z and (b) the child scores Z' induced by a split, the SCAFF gain SG associated with that split is defined as:

$$SG = \text{SCAFF}(Z', Y, S, \Theta) - \text{SCAFF}(Z, Y, S, \Theta). \tag{5.10}$$

The split with maximal SG across all evaluated splits is selected if and only if its corresponding $SG \geq 0$. Otherwise, no splitting occurs.

SCAFF is not only able to handle binary sensitive attributes but also extends to the multivariate and multicategorical scenarios, including intersectional factors (i.e., the combination of sensitive attributes) [Buolamwini and Gebru, 2018] via a one-versus-rest (OvR) approach [Tax and Duin, 2002]. The AUC_S used in SCAFF is the maximum OvR, since no sensitive attribute should have priority over fairness.

An example of SCAFF evaluation can be viewed in Fig. 5.2, in which the OvR $\text{AUC}_\text{S} = \max(0.6, 0.917) = 0.917$. In the aforementioned example, we mention that Z scores are given as $P(Y_+)$ in a node. We remark that our methods extends trivially to the bagging (i.e., random forest) case by considering the final score of a sample as the average score across all trees. Yet, other Z score definitions are viable; e.g., (gradient) boosting techniques compute Z by iteratively updating existing sample scores [Hastie et al., 2009]. In that sense, samples within the same child node may have distinct Z scores.

Our method extends to such boosting cases since SG relies on Z, regardless of its computation. In contrast, traditional tree learning algorithms do not extend to boosting, since no Z scores are incorporated into the splitting criteria. We remark that, for samples in the same node which have distinct Z scores, the computation of the AUC must follow the traditional approach (Eq. 5.5).

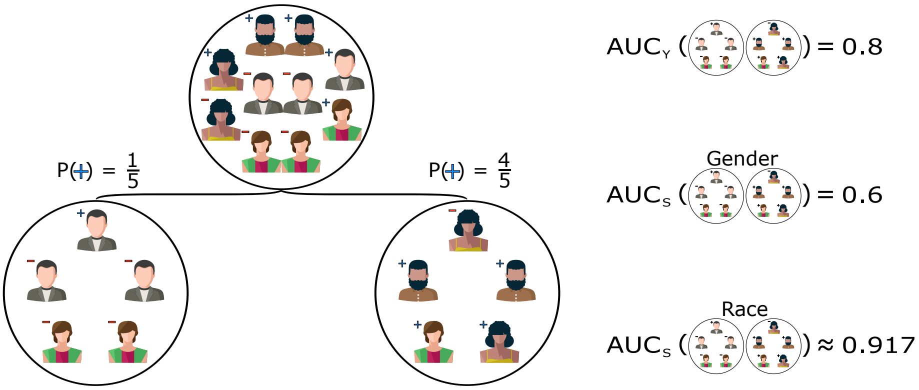


Figure 5.2: **Computing AUC values for SCAFF.** AUC_Y and AUC_S in a system with 10 samples, a class label, and two sensitive attributes (gender and race).

5.5 Experiments

For the description of our experiments, we begin by mentioning the datasets and how we used them (Section 5.5.1); we then characterise the experimental setup deployed to (1) gather the performance and fairness values and (2) report on the relationship between the threshold-independent and threshold-dependent demographic parities (Section 5.5.2). We compared SCAFF against other fair splitting criteria by using benchmark fairness datasets. Since the methods against which we compare our approach are neither suited for multivariate nor multicategorical sensitive attributes, we focus on the single binary sensitive attribute case first. We additionally experimented on a single dataset to explore how SCAFF handles multiple sensitive attributes simultaneously as well as multicategorical values. Lastly, we tested the quantitative relationship of the strong demographic parity yielded by our method with the corresponding demographic parity at different decision-thresholds. For reproducibility, our experiments are made available in [Pereira Barata, 2021].

5.5.1 Datasets

Three binary classification datasets were used. These are benchmark datasets used for fairness methods [Quy et al., 2021]. Each of them has at least one sensitive attribute. Specifically, we employed the following: (a) *Bank* ($45,211$ instances, 50 features) in which the sensitive attribute is the binary condition of age ≥ 65 (b) *Adult* ($45,222$ instances, 97 features), where the sensitive attribute may be either (i) race $\in \{\text{white}, \text{non-white}\}$ or (ii) gender $\in \{\text{male}, \text{female}\}$; and (c) *Recidivism* ($6150$ instances, 8 features) of which the sensitive attributes may be either (i) race $\in \{\text{white}, \text{non-white}\}$ or (ii) gender $\in \{\text{male}, \text{female}\}$.

For the binary sensitive attribute case, we considered each dataset-sensitive attribute configuration, making for a total of five different dataset configurations. Two scenarios were further set in which the *Adult* dataset was considered: (i) the multiple sensitive attribute scenario such that both sensitive attributes (race and gender) were handled simultaneously; and (ii) the multicategorical sensitive attribute scenario in which the intersectional attributes $\{$non-white female (NWF), non-white male (NWM), white female (WF), white male (WM)$\}$ were concurrently considered.

5.5.2 Experimental Setup

To provide an adequate comparison between our splitting criterion and the state-of-the-art, we considered previous works in fair splitting criteria; specifically, the works proposed by [Kamiran et al., 2010] and [Zhang and Ntoutsi, 2019]. For each dataset configuration, and for all methods, the same 10-fold CV was applied.

To measure classification performance and algorithm fairness, $\mathrm{AUC_Y}$ (the accepted standard measure for classifier performance) and $\mathrm{AUC_S}$ were used, respectively. The performance and fairness measures across test folds were averaged to produce a single value pair for each dataset, per method, and in our case for each value of orthogonality Θ. For all methods, the classification output scores Z of samples were computed as the $P(Y_+)$ of the terminal leaf node of a single tree, as previously shown in Fig. 5.2.

To be able to achieve state-of-the-art performance, each method was deployed as a random forest (i.e., bagging) [Breiman, 2001]. As such, the final classification score of a sample is the average Z model output of all terminal nodes across the different trees generated. Throughout all methods, the same set of hyperparameters was used, such as the number of trees (500), the maximum depth of each tree (4), and the random seed initialisation.

Bootstrapping, random feature selection, and continuous-feature discretisation were also applied, given their prevalence in real-world implementations of tree-based algorithms, such as [Chen and Guestrin, 2016]. For our method, a range of 11 values for Θ was used between 0 and 1. For details of the implementation, see [Pereira Barata, 2021].

To relate the threshold-dependent and threshold-independent demographic parities, decision thresholds were applied to the classifier outputs of our method across different values of Θ for the different datasets. The thresholds were considered as 9 quantiles values between 0.1 and 0.9 of each test set output and, consequently, demographic parity —defined in Section 5.3.1— was averaged over folds.

We measured, at each decision threshold —along Θ values— the Pearson correlation coefficient [Kirch, 2008], and the respective null hypothesis p-values, between strong demographic parity (as AUC_S) and demographic parity. The purpose is to check whether the behaviour of strong demographic parity across Θ transfers to that of the induced demographic parity.

5.6 Results

In this section, we present the results of our experiments. We report on the classification performance, fairness, and tunability of the performance-fairness trade-off achieved by our method via orthogonality Θ. We do so for the aforementioned sensitive attribute configurations: binary (Section 5.6.1), and non-binary (Section 5.6.2). Specifically for the binary configuration, we compare our method to the competing approaches. Finally, we show how the strong demographic parity (measured in AUC_S) yielded by our method translates to the induced demographic parity across different (a) decision thresholds and (b) values of orthogonality Θ (Section 5.6.3).

5.6.1 Binary Sensitive Attribute

To regard the performance and fairness of all methods per dataset configuration, see Fig. 5.3. For our method, each point corresponds to a value of $\Theta \in [0, 1]$. An orthogonality value $\Theta = 0$ is equivalent to a traditional classifier and corresponds to the right-most point. Conversely, $\Theta = 1$ corresponds to the left-most point. In the horizontal axis, AUC_S represents (un)fairness. The vertical axis depicts AUC_Y as classification performance.

Unlike the other methods which output a single performance-fairness value (represented as a point), our SCAFF method produces a performance-fairness trade-off curve along Θ. This is advantageous as it provides a way for practitioners to make informed decisions. The impact of Θ on the tunability of the performance-fairness trade-off for each dataset-sensitive attribute pair is consistent: as increasingly greater values of Θ are used, the greater the fairness and lesser the classification performance.

Noticeably, in *Bank (Age)*, SCAFF was able to reduce AUC_S by 0.2 at a loss in performance of only 0.02. Overall, our method consistently performs better in the combination of classification performance and fairness, allowing for a suitable target point. It is a convincing result of (1) the use of AUC in the splitting criterion and (2) the flexibility of the orthogonality parameter Θ.

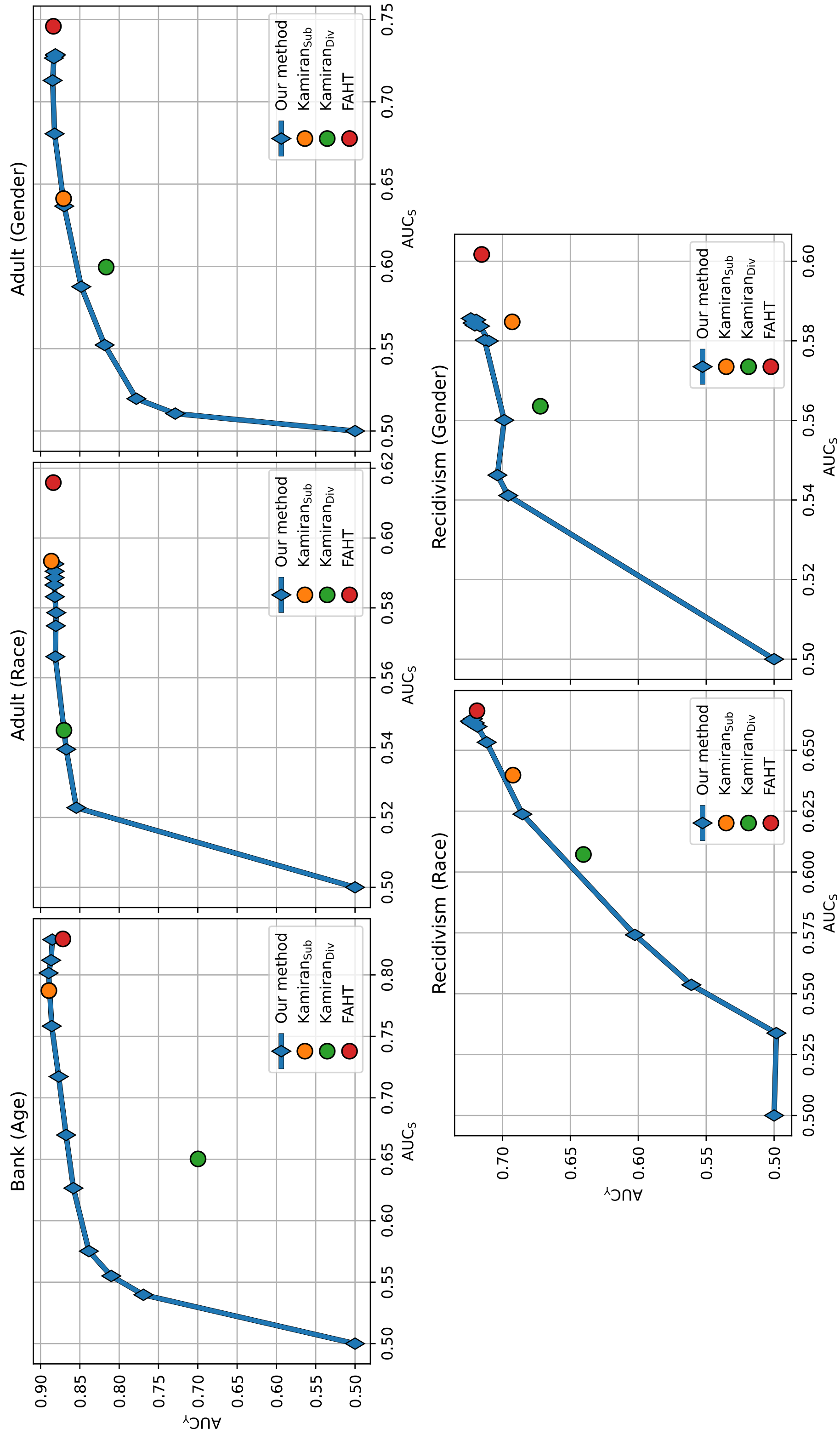


Figure 5.3: **Performance-fairness across methods per dataset.** Measures of AUC_S (horizontal axis) and AUC_Y (vertical axis) are shown. For our method, each point from right to left indicates a value of $\Theta \in [0, 1]$ in increasing order.

5.6.2 Multiple and Multicategorical Cases

We present in Fig. 5.4 the outcomes of the dataset configurations for multiple sensitive attributes —*Adult (Multiple)* in the left panel — and multicategorical sensitive attribute values, considered as the intersectional values: *Adult (Intersectional)* in the right panel.

For both panels, across different values of orthogonality Θ (horizontal axis), the classification performance $\mathrm{AUC_Y}$ is shown in blue and the different $\mathrm{AUC_S}$ are provided (vertical axis). To the left, the $\mathrm{AUC_S}$ for race and gender can be regarded; to the right, the $\mathrm{AUC_S}$ for each of the different intersectional sensitive attribute values are displayed: NWF, NWM, WF, and WM.

Focusing on the *Adult (Multiple)* configuration, it is witnessable that the behaviour of the fairness measures along Θ match those of the *Adult (Race)* and *Adult (Gender)* previously shown in Fig. 5.3: greater values of orthogonality translate to greater values of fairness (decreasing $\mathrm{AUC_S}$) and lesser classification performance $\mathrm{AUC_Y}$. This is expected, since the performance-fairness tradeoff phenomenon is known.

SCAFF was able to reduce the bias towards both sensitive attributes simultaneously whilst maintaining adequate classification performance; in particular at $\Theta = 0.7$, both race and gender $\mathrm{AUC_S} = 0.55$ (a remarkably low bias value), and $\mathrm{AUC_Y}$ is above 0.8 indicating model adequacy. Similarly for *Adult (Intersectional)* at the same value of the orthogonality parameter $\Theta = 0.7$, our method was able to converge the bias of all sensitive attribute values to sensible values concurrently whilst maintaining proper classification performance.

These results show our proposed method is able to produce adequate classification models with regards to multiple and multicategorical sensitive attributes which maximise performance with the least decrease in fairness. To put it differently, our method is able to exploit the performance-fairness tradeoff even for multiple and multicategorical sensitive attributes.

We remark, however, one limitation of our OvR approach. Since the OvR $\mathrm{AUC_S}$ along multiple attributes or values is evaluated as its maximum (as described in Section 5.4.3), there is no guarantee that all attributes will have their biases decreased along Θ: regard the *slight* increase in NWM bias.

Yet, this characteristic of our approach inherently bounds the highest possible value of bias. In other words, along Θ, the maximum value of $\mathrm{AUC_S}$ is strictly monotonically decreasing. The remark is further corroborated by the NWF, WF, and WM intersectional sensitive attributes, of which the curves behave in a nearly-identical manner along the different values of Θ. Under the assumption that none of the sensitive attributes is of greater importance than any other, the maximally-valued sensitive attribute should always be considered as the attribute by which fairness is measured.

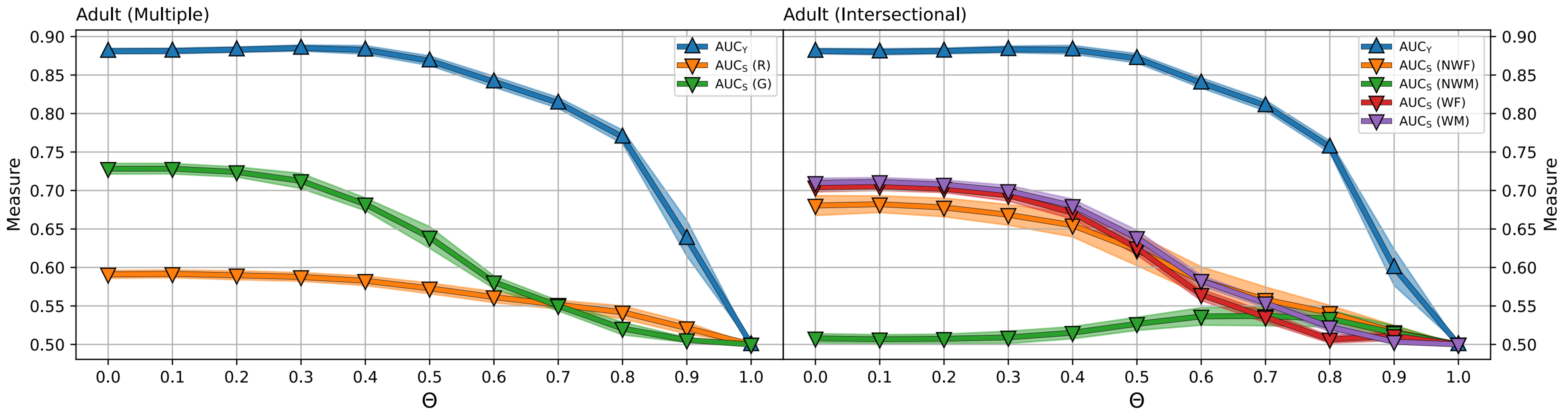


Figure 5.4: **Performance-fairness for the multiple and intersectional cases.** Across different values of orthogonality Θ (horizontal axis), average and standard deviation of AUC$_Y$ and AUC$_S$ values (vertical axis) are shown for the *Adult* dataset.

5.6.3 Relationship with Demographic Parity

Below, we describe the results of applying our method to the five dataset configurations for different values of Θ, and measuring the corresponding (threshold-dependent) demographic parity at different decision thresholds. The purpose is to determine if (1) threshold-independence extends across arbitrary decision thresholds, and (2) if changes in Θ induce an equivalent behaviour between demographic parity and strong demographic parity.

In Fig. 5.5, it is shown how for different decision thresholds (horizontal axis), the mean demographic parity (vertical axis) —across all test folds— behaves with different values of Θ (differently-coloured lines), for the five binary sensitive attribute dataset configurations. An additional panel is provided (bottom-right), where for each value of Θ (horizontal axis), the variation of demographic parity across decision thresholds for each dataset is present.

Across all dataset configurations, and particularly noticeable in those with high demographic parity —concretely *Bank (Age) and Adult (Gender)*— the effect of the orthogonality parameter Θ is generally the same: as orthogonality values increase, values for demographic parity decrease, regardless of the decision threshold selected.

The spread of demographic parity (measured as standard deviation) also decreases along Θ, for different decision thresholds. To put it differently, higher values of Θ translate to greater threshold-independence. This is sensical since, by definition, SCAFF directly optimises for threshold-independent measures.

To grasp the relationship between strong and threshold-dependent demographic parities, regard Table 5.1. Each row depicts a decision threshold at which demographic parity was computed; a column indicates a dataset configuration. A cell depicts the Pearson correlation coefficient between the two measures of fairness along the parameter Θ, for the decision threshold. The coefficients represent how similar the behaviour between threshold-dependent and - independent demographic parities is, induced by shifts in Θ.

Noteworthily, bolded entries indicate a statistical significance of $\alpha = 0.05$ towards the null hypothesis of no correlation. Safe for a single outlying entry —threshold 0.9 in the *Adult (Race)* configuration, in which the value of demographic parity is negligible— all table entries are consistently high and of statistical significance. This shows that the effect of shifting the orthogonality parameter Θ is, in practice, identical for both types of demographic parity regardless of the decision threshold selected, validating our method with respect to threshold independence.

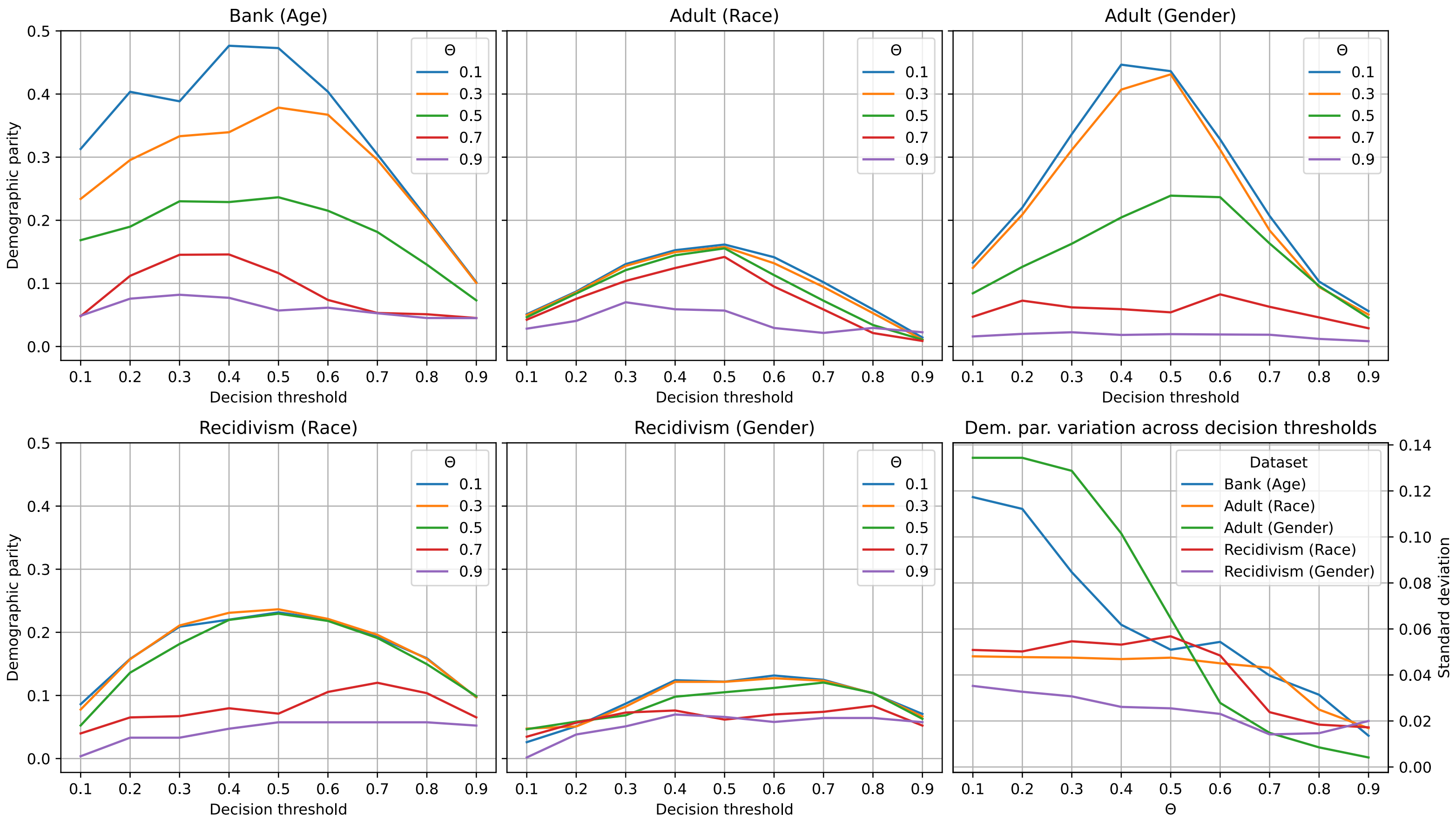


Figure 5.5: **Effect of orthogonality over demographic parity.** Decision thresholds are shown in the horizontal axis. Values of demographic parity are represented in the vertical axis. Values of Θ are highlighted with different colours. Bottom-right panel: standard deviation (vertical axis) of demographic parity across thresholds and Θ (horizontal axis).

Table 5.1: Pearson correlation coefficients between strong demographic parity (measured as AUC_S) and demographic parity, along Θ, for different decision thresholds in the five dataset configurations; bolded entries indicate a null hypothesis p-value ≤ 0.05.

	Dataset				
Th	Bank (A)	Adult (R)	Adult (G)	Recid. (R)	Recid. (G)
0.1	**0.983**	**0.963**	**0.994**	**0.937**	**0.839**
0.2	**0.984**	**0.965**	**0.997**	**0.995**	**0.895**
0.3	**0.993**	**0.971**	**0.994**	**0.987**	**0.968**
0.4	**0.988**	**0.992**	**0.991**	**0.995**	**0.949**
0.5	**0.997**	**0.988**	**0.995**	**0.990**	**0.973**
0.6	**0.993**	**0.994**	**0.995**	**0.998**	**0.975**
0.7	**0.984**	**0.979**	**0.984**	**0.991**	**0.992**
0.8	**0.975**	**0.871**	**0.919**	**0.983**	**0.984**
0.9	**0.941**	0.267	**0.947**	**0.944**	**0.922**

5.7 Chapter Conclusion

In the present work, we introduced SCAFF. By doing so, we proposed a learning algorithm which simultaneously (1) optimises for threshold-independent performance —AUC— and fairness —strong demographic parity— (2) is able to handle various multicategorical sensitive attributes simultaneously, (3) is tunable with respect to the performance-fairness trade-off via an orthogonality parameter Θ, and (4) easily extends to bagging and (gradient) boosting.

Moreover, we empirically validated our method through experimentation on benchmark datasets traditionally used in the fairness literature. Then we validated our experiments with real datasets. Here, we showed that our approach outperformed the competing state-of-the-art criteria methods, by its predictive performance and model fairness, as well as by its capability of handling multiple sensitive attributes simultaneously, of which the values may be valued multicategorically. Moreover, we demonstrated how the behaviour of strong demographic parity induced by our method extends to the threshold-dependent demographic parity.

As future work, we recommend to extend the current framework from learning classification problems towards other learning paradigms. Ultimately, the development and deployment of fair machine learning approaches within sensitive domains is the goal in this field of research.

Chapter 6

Conclusions

In the final chapter, we answer the three RQs in Section 6.1. Then, we address the problem statement and clearly identify the research results together with their conclusions in Section 6.2. Lastly, in Section 6.3, research directions are proposed with the intent of furthering reliable and fair data-driven risk assessment applications.

6.1 Answers to the Research Questions

Below, the RQs are reiterated as formulated in Chapter 1. Each question is answered separately.

> **RQ1:** *Given data with missing values, which (a) missing data-handling technique and (b) learning algorithm should be jointly selected such that, regardless of the missing mechanism, the detriment to the downstream task performance is minimal when compared to the non-missing (unavailable) case?*

When dealing with real-world data, a non-MCAR scenario is traditionally assumed. Thus, a viable option is to use the missing-indicator method, encoding the missingness itself [Lipton et al., 2016]. However, this assumption does not always hold; in such cases, the missing-indicator method may even deteriorate the performance of the (downstream) classification task, when compared to an imputation approach.

It is established in Section 2.6 that, under MCAR, the differences in the downstream classification task performance between (a) imputation, and (b) missing-indicator are negligible, if the appropriate learner is used. Specifically via feature selection protocols, it is possible to learn classification models of which the performances are statistically *indistinguishable* across the two different methods used to handle missingness.

Given that real-world data are seldom MCAR and that, even under the MCAR mechanism, the performance decrease can be made indistinguishable between imputation and missing-indicator, the answer to RQ1 is, therefore, that the missing-indicator method, in conjunction with decision tree-based learners —particularly via gradient boosting— is an adequate solution. In doing so, the detriment to classification performance should be minimal, whether under MCAR or non-MCAR.

One limitation to our answer is that, in our experimental design, both training and test sets are identically distributed with respect to the missing mechanism. Further studies could be conducted to assess classifier performance under a *slightly* different, yet most impactful scenario, in which the training set (i.e., the available training data) and test set (representing the model deployment) are *differently* distributed with respect to the mechanism of missingness. For example, if a classifier were to be learned from data under MCAR, coupled with a missing-data handling technique, what would the expected performance be if the test data were either *non-missing* or non-MCAR?

> **RQ2:** *Given data with label noise, how can noisy-samples be (a) adequately detected, and (b) used to learn a well-performing model?*

In Chapter 3, the notion of *crosslier* is introduced. Crossliers are anomalous instances with respect to a domain-sensible category. These instances may be misconducts as their characteristics position them farther from their category cluster, across the decision boundary towards one or more other categories. With respect to the Inspectorate, waste category crossliers were presented as potential misconducts.

By learning well-calibrated probabilistic classifiers, it is possible to use the output probabilities of trained models towards a set of new samples, where lowest posterior probability indicates a most probable crosslier. To achieve this, we propose the EXPOSE method. Our method addresses the entire dataset in a CV manner, generating well-calibrated sample probabilities for the test sets. As a result, all samples have comparable crosslier scores with which crosslier detection may be performed. This process answers RQ2(a).

Yet, we denote that (as established in Chapter 1), data within the Inspectorate represent the *administrative* reality. What might be considered misconduct, could in fact be simply a data quality issue (e.g., an entry error). The distinction is, however, dependent on which reality the data represents. For this reason, special care is required when considering the real-world *meaning* of a sample with a high crosslier score. Ultimately, a case-by-case approach should be taken if deployment is to be reliable in the Inspectorate.

The amount of noise in observations can be quantified so long as there is access to the posterior probabilities of those samples. After quantification using the EXPOSE method, the probabilities may be manipulated into weights reflective of the *clean* target distribution. In Chapter 4, we proposed a viable weighting scheme in which we consider the weight of an observation as the log-odds of its posterior probability. This proposed learning method is coined DENOISE. We showed that the resulting models learned on noisy data with this sample weighting scheme are well-performing. This result answers RQ2(b).

Although we are confident on the performance of our method, we must denote that the manner by which noise was artificially generated is a limitation: instead of a random univariate approach to noise generation, a more complex approach could have been deployed. For example, selecting features of which the importance for the final classifier is high. Further studies should be conducted, where learning performance could be assessed not only in terms of noise probability, but also in terms of the generation of noise itself.

> **RQ3:** *How can we, from biased data, learn a model tunable with respect to the performance-fairness trade-off such that the selection of the trade-off point is made intuitive for the relevant stake-holders?*

In Chapter 5, a decision tree learning framework is leveraged by incorporating a threshold-agnostic fair classification splitting criterion termed SCAFF. The splitting criterion is formulated as a weighted linear combination of (a) the AUC towards the class label and (b) the strong demographic parity, implemented as the AUC towards the sensitive attribute. To provide tunability with respect to performance-fairness trade-off, an orthogonality parameter $\Theta \in [0, 1,]$ is part of the splitting criterion.

By analysing the performance-fairness trade-off curve, an appropriate value of Θ can be selected according to the application domain requirement(s). In addition, multiple multicategorical sensitive attributes may be addressed simultaneously by minimising the maximal sensitive AUC across all sensitive attributes as the term in the splitting criterion. Through experimentation and comparison with other fair splitting criteria, we validated our method for various datasets and sensitive attribute scenarios. Our proposed SCAFF method, specifically via the orthogonality parameter Θ, answer RQ3.

While the implementation of our method easily extends to bagging and boosting frameworks, the computational costs associated with either of the extensions are not equivalent. Concretely, the computation of AUC in the boosting framework needs to follow the traditional and more time complex approach: unlike with bagging where each tree node can be represented as either a positive or negative class prediction, nodes in boosted trees contain samples each with their sample-specific prediction score.

6.2 Answer to the Problem Statement

We may now give an answer to the PS based on the answers to the RQs provided above.

> **PS:** *How can machine learning methods advance data-driven risk assessment by the Inspectorate in a reliable and fair manner?*

In our research, we focused on two aspects inherent to the development and deployment of high-risk AI: reliability and fairness. In particular, we highlighted classification models towards risk assessment under the current EU movement towards trustworthy AI. In line with our research, we first address reliability and thereafter fairness.

At the start, we transposed the aspect of reliability into that of two important data quality issues: *missingness* and *noise*. For *missingness*, as described in Chapter 2, we experimented with different methods which handle missing data, by addressing RQ1. Based on our results, we may conclude that with real-world data, a missing-indicator method in conjunction with a decision tree-based learner is a viable solution to address the problem of missing data.

With respect to *noise*, discussed in Chapter 3, we considered *noise in data* as potential real-world misconduct. Hence, we proposed a method to detect it by addressing RQ2(a). Moreover in Chapter 4 we leveraged this noise detection towards a novel method of learning adequately-performing models from noisy data by addressing RQ2(b).

With respect to *fairness*, we claim that countering biased data is crucial in risk assessment. In Chapter 5, we consolidated this claim by answering RQ3. Our main result is proposing a decision tree learning algorithm which takes into account bias in data to produce a classifier that performs adequately and is easily-tunable in terms of the performance-fairness trade-off.

We contribute to reliable and fair machine learning methods for risk assessment by the Inspectorate via Chapters 2– 5. On top of that, we reinforce the principle given in Chapter 1 towards trustworthy AI: *the expertise of domain experts must not be replaced by automation*, but rather *enhanced* by it. To note, while we state that we contribute towards reliability and fairness, we wish to make very explicit that the road to data-driven solutions for the problem of risk assessment has merely started being paved.

The opportunities for continuation are ample, and present their own set of stimulating challenges: not only in terms of the actual implementation of the solutions presented in this thesis into the daily operations of the Inspectorate, but also in terms of the myriad of other data-related issues that this thesis did not cover. As such, we follow with suggestions for potential future research in this domain.

6.3 Future Research

Dealing with real-world data in the inspection domain is (1) a sensitive and laborious undertaking and (2) a stimulating research area. One straightforwards research direction is to procure a joint solution to the individual issues derived from the three investigations (RQs 1–3). To put it differently, it would be advantageous to develop learners which are able to simultaneously address (1) missingness, (2) noise, and (3) fairness, while remaining highly performing. This could be achieved by applying the missing-indicator method, in conjunction with the incorporation of our sample weighing scheme into our fair tree learner.

Another prominent research direction, specifically with application in the Inspectorate, may be to broaden the scope of how data is represented, prior to learning. In other words, while the classification problems remain the same (e.g., identifying misconduct in different sub-fields), there is still opportunity for enhancing the feature representation/embedding process which, in turn, might promote superior model performance. A common approach to adequately model the complex interactions between individuals in a dataset, builds on concepts from the relatively young field of network science [Barabási, 2016]. It is widely accepted in the literature that networks are the *de facto* data architectures towards modelling the behaviour and dynamics of real-world systems, as further corroborated by our most recent work towards automated and fair ship targetting [de Bruin et al., 2022]. Therefore, we believe that more applications should be tractable following such approaches.

Lastly, another problem of relevance to the Inspectorate relates to the following. By definition, historical inspection data is neither an independent nor identically distributed sample of the entire population. In other words, since the function of the inspectors of the ILT is to select for inspection the cases which are of highest risk, the selection will generate data samples of which the distribution in feature space is not representative of the entire pool of cases. This makes it difficult to learn a classifier that distinguishes between more or less risky samples adequately due to this under-representation of feature space, which may lead to inspection blind-spots; i.e., regions in feature space which are considered by the inspectors and, consequently, the classifier to be of non-interest when in fact they pertain to risk behaviour. One way to address this issue would be to deploy active sampling methods such that, rather than targetting samples of which the true (unseen) label most probably indicates risk, samples would be selected towards increasing the generalisation of a given model. Another approach would be to leverage the study of co-domain adaptation (or covariate shift), under which the assumption is that the training and deployment distributions are different between each other.

Ultimately, our goal is to enact tangible change in the way the Inspectorate operates. For this purpose, however, action by the responsible agents, and not solely the machine learners, is required. The shift towards a data-driven Inspectorate has only just begun.

References

[Abellán and Masegosa, 2010] Abellán, J. and Masegosa, A. R. (2010). Bagging decision trees on data sets with classification noise. In *International Symposium on Foundations of Information and Knowledge Systems*, pages 248–265. Springer.

[Abellán and Moral, 2003] Abellán, J. and Moral, S. (2003). Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, 18(12):1215–1225.

[Ahirwar, 2020] Ahirwar, J. P. (2020). Five laws of library science and information economics. *Informatics Studies*, 7(1).

[Alazzam et al., 2019] Alazzam, H., Alsmady, A., and Shorman, A. A. (2019). Supervised detection of IoT botnet attacks. In *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems*, pages 1–6.

[Alcalá-Fdez et al., 2009] Alcalá-Fdez, J., Sanchez, L., Garcia, S., del Jesus, M. J., Ventura, S., Garrell, J. M., Otero, J., Romero, C., Bacardit, J., Rivas, V. M., et al. (2009). KEEL: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13(3):307–318.

[Amer and Goldstein, 2012] Amer, M. and Goldstein, M. (2012). Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer. In *Proceedings of the 3rd RapidMiner Community Meeting and Conference (RCOMM 2012)*, pages 1–12.

[Amiri and Jensen, 2016] Amiri, M. and Jensen, R. (2016). Missing data imputation using fuzzy-rough methods. *Neurocomputing*, 205:152–164.

[Angenent et al., 2020] Angenent, M. N., Pereira Barata, A., and Takes, F. W. (2020). Large-scale machine learning for business sector prediction. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 1143–1146.

[Angluin and Laird, 1988] Angluin, D. and Laird, P. (1988). Learning from noisy examples. *Machine Learning*, 2(4):343–370.

[Asuncion and Newman, 2007] Asuncion, A. and Newman, D. (2007). UCI machine learning repository.

[Azar and El-Metwally, 2013] Azar, A. T. and El-Metwally, S. M. (2013). Decision tree classifiers for automated medical diagnosis. *Neural Computing and Applications*, 23(7):2387–2403.

[Barabási, 2016] Barabási, A. L. (2016). *Network Science*. Cambridge University Press.

[Barocas et al., 2017] Barocas, S., Hardt, M., and Narayanan, A. (2017). Fairness in machine learning. *Nips tutorial*, 1:2.

[Bartlett et al., 2006] Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156.

[Beale and Little, 1975] Beale, E. M. and Little, R. J. (1975). Missing values in multivariate analysis. *Journal of the Royal Statistical Society: Series B (Methodological)*, 37(1):129–145.

[Bechavod and Ligett, 2017] Bechavod, Y. and Ligett, K. (2017). Penalizing unfairness in binary classification. *arXiv preprint arXiv:1707.00044*.

[Beigman Klebanov and Beigman, 2009] Beigman Klebanov, B. and Beigman, E. (2009). From annotator agreement to noise models. *Computational Linguistics*, 35(4):495–503.

[Bennett, 2001] Bennett, D. A. (2001). How can I deal with missing data in my study? *Australian and New Zealand Journal of Public Health*, 25(5):464–469.

[Bennett, 2000] Bennett, P. N. (2000). Assessing the calibration of naive bayes posterior estimates. Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science.

[Bergstra and Bengio, 2012] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2).

[Bertsimas et al., 2017] Bertsimas, D., Pawlowski, C., and Zhuo, Y. D. (2017). From predictive methods to missing data imputation: an optimization approach. *The Journal of Machine Learning Research*, 18(1):7133–7171.

[Böken, 2021] Böken, B. (2021). On the appropriateness of Platt scaling in classifier calibration. *Information Systems*, 95:101641.

[Bonner et al., 2015] Bonner, S., McGough, A. S., Kureshi, I., Brennan, J., Theodoropoulos, G., Moss, L., Corsar, D., and Antoniou, G. (2015). Data quality assessment and anomaly detection via map/reduce and linked data: a case study in the medical domain. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 737–746. IEEE.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[Breiman et al., 1984] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth.

[Breunig et al., 2000] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 93–104.

[Brink et al., 2016] Brink, H., Richards, J., and Fetherolf, M. (2016). *Real-world machine learning*. Simon and Schuster.

[Brodley and Friedl, 1996] Brodley, C. E. and Friedl, M. A. (1996). Improving automated land cover mapping by identifying and eliminating mislabeled observations from training data. In *IGARSS'96, 1996 International Geoscience and Remote Sensing Symposium*, volume 2, pages 1379–1381. IEEE.

[Brodley and Friedl, 1999] Brodley, C. E. and Friedl, M. A. (1999). Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167.

[Buolamwini and Gebru, 2018] Buolamwini, J. and Gebru, T. (2018). Gender shades: intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91. PMLR.

[Cariou and Wolff, 2011] Cariou, P. and Wolff, F.-C. (2011). Do port state control inspections influence flag-and class-hopping phenomena in shipping? *Journal of Transport Economics and Policy (JTEP)*, 45(2):155–177.

[Carpenter and Kenward, 2012] Carpenter, J. and Kenward, M. (2012). *Multiple Imputation and its Application*. John Wiley & Sons.

[Chan and Treleaven, 2015] Chan, S. and Treleaven, P. (2015). Continuous model selection for large-scale recommender systems. In *Handbook of Statistics*, volume 33, pages 107–124. Elsevier.

[Chapelle et al., 2006] Chapelle, O., Scholkopf, B., and Zien, A. (2006). Semi-supervised learning. 2006. *Cambridge, Massachusettes: The MIT Press View Article*, 2.

[Chen et al., 2014] Chen, M., Yang, C., Li, C., Hou, L., Chen, X., and Zhao, H. (2014). Admixture mapping analysis in the context of GWAS with GAW18 data. In *BMC Proceedings*, pages 1–5. BioMed Central.

[Chen and Guestrin, 2016] Chen, T. and Guestrin, C. (2016). XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

[Cho et al., 2020] Cho, J., Hwang, G., and Suh, C. (2020). A fair classifier using kernel density estimation. *Advances in Neural Information Processing Systems*, 33:15088–15099.

[Choi et al., 2019] Choi, J., Dekkers, O. M., and le Cessie, S. (2019). A comparison of different methods to handle missing data in the context of propensity score analysis. *European Journal of Epidemiology*, 34(1):23–36.

[Choudhary and Gianey, 2017] Choudhary, R. and Gianey, H. K. (2017). Comprehensive review on supervised machine learning algorithms. In *2017 International Conference on Machine Learning and Data Science (MLDS)*, pages 37–43. IEEE.

[Chu et al., 2013] Chu, X., Ilyas, I. F., and Papotti, P. (2013). Discovering denial constraints. *Proceedings of the VLDB Endowment*, 6(13):1498–1509.

[Claesen and De Moor, 2015] Claesen, M. and De Moor, B. (2015). Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*.

[CLAIRE, 2021] CLAIRE (2021). Response to the European Commission's proposal for AI regulation and 2021 coordinated plan on AI. `https://claire-ai.org/wp-content/uploads/2021/08/CLAIRE-EC-AI-Regulation-Feedback.pdf`.

[Corbett-Davies and Goel, 2018] Corbett-Davies, S. and Goel, S. (2018). The measure and mismeasure of fairness: a critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*.

[Corbett-Davies et al., 2017] Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., and Huq, A. (2017). Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–806.

[Cuendet et al., 2007] Cuendet, S., Hakkani-Tür, D., and Shriberg, E. (2007). Automatic labeling inconsistencies detection and correction for sentence unit segmentation in conversational speech. In *International Workshop on Machine Learning for Multimodal Interaction*, pages 144–155. Springer.

[Dardanoni et al., 2011] Dardanoni, V., Modica, S., and Peracchi, F. (2011). Regression with imputed covariates: a generalized missing-indicator approach. *Journal of Econometrics*, 162(2):362–368.

[Dastian, 2018] Dastian, J. (2018). Amazon scraps secret ai recruiting tool that showed bias against women. *Reuters*.

[de Bruin et al., 2022] de Bruin, G. J., Pereira Barata, A., van den Herik, H. J., Takes, F. W., and Veenman, C. J. (2022). Fair automated assessment of non-compliance in cargo ship networks. *EPJ Data Science*, 11(1):13.

[Diekmann and Jann, 2010] Diekmann, A. and Jann, B. (2010). Benford's law and fraud detection: facts and legends. *German Economic Review*, 11(3):397–401.

[Dietterich, 2000] Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157.

[Ding and Simonoff, 2010] Ding, Y. and Simonoff, J. S. (2010). An investigation of missing data methods for classification trees applied to binary response data. *Journal of Machine Learning Research*, 11(1).

[Dogru and Subasi, 2018] Dogru, N. and Subasi, A. (2018). Traffic accident detection using random forest classifier. In *2018 15th Learning and Technology Conference (L&T)*, pages 40–45. IEEE.

[Dressel and Farid, 2018] Dressel, J. and Farid, H. (2018). The accuracy, fairness, and limits of predicting recidivism. *Science Advances*, 4(1):eaao5580.

[Dwork et al., 2012] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226.

[Enders, 2010] Enders, C. K. (2010). *Applied Missing Data Analysis*. Guilford Press.

[European Commission, 2018a] European Commission (2018a). Waste classification. `https://ec.europa.eu/environment/waste/framework/list.htm`.

[European Commission, 2018b] European Commission (2018b). Waste legislation. `https://ec.europa.eu/environment/waste/legislation`.

[European Commission, 2019a] European Commission (2019a). A definition of AI: main capabilities and disciplines. `https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=56341`.

[European Commission, 2019b] European Commission (2019b). Ethics guidelines for trustworthy AI. `https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=60419`.

[European Commission, 2019c] European Commission (2019c). Proposal for a regulation on a European approach for artificial intelligence. `https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-european-approach-artificial-intelligence`.

[European Commission, 2021] European Commission (2021). Regulatory framework proposal on artificial intelligence. `https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai`.

[Feldman et al., 2015] Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. (2015). Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268.

[Feng et al., 2011] Feng, X., Wu, S., and Liu, Y. (2011). Imputing missing values for mixed numeric and categorical attributes based on incomplete data hierarchical clustering. In *International Conference on Knowledge Science, Engineering and Management*, pages 414–424. Springer.

[Flach, 2016] Flach, P. A. (2016). ROC analysis. In *Encyclopedia of Machine Learning and Data Mining*, pages 1–8. Springer.

[Frénay and Verleysen, 2013] Frénay, B. and Verleysen, M. (2013). Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869.

[Frery et al., 2017] Frery, J., Habrard, A., Sebban, M., Caelen, O., and He-Guelton, L. (2017). Efficient top rank optimization with gradient boosting for supervised anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 20–35. Springer.

[Friedman, 2001] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

[Fürber, 2016] Fürber, C. (2016). Semantic technologies. In *Data Quality Management with Semantic Technologies*, pages 56–68. Springer.

[Gamberger et al., 1996] Gamberger, D., Lavrač, N., and Džeroski, S. (1996). Noise elimination in inductive concept learning: a case study in medical diagnosis. In *International Workshop on Algorithmic Learning Theory*, pages 199–212. Springer.

[Gamberger et al., 1999] Gamberger, D., Lavrac, N., and Groselj, C. (1999). Experiments with noise filtering in a medical domain. In *ICML*, volume 99, pages 143–151.

[García-Laencina et al., 2015] García-Laencina, P. J., Abreu, P. H., Abreu, M. H., and Afonoso, N. (2015). Missing data imputation on the 5-year survival prediction of breast cancer patients with unknown discrete values. *Computers in Biology and Medicine*, 59:125–133.

[García-Laencina et al., 2010] García-Laencina, P. J., Sancho-Gómez, J.-L., and Figueiras-Vidal, A. R. (2010). Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282.

[Garciarena and Santana, 2017] Garciarena, U. and Santana, R. (2017). An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers. *Expert Systems with Applications*, 89:52–65.

[George and Vidyapeetham, 2012] George, A. and Vidyapeetham, A. (2012). Anomaly detection based on machine learning: dimensionality reduction using PCA and classification using SVM. *International Journal of Computer Applications*, 47(21):5–8.

[Ghosh et al., 2017] Ghosh, A., Kumar, H., and Sastry, P. (2017). Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[Goddard, 2017] Goddard, M. (2017). The EU general data protection regulation (GDPR): European regulation that has a global impact. *International Journal of Market Research*, 59(6):703–705.

[Goh et al., 2016] Goh, G., Cotter, A., Gupta, M., and Friedlander, M. (2016). Satisfying real-world goals with dataset constraints. *arXiv preprint arXiv:1606.07558*.

[Goldfarb-Tarrant et al., 2020] Goldfarb-Tarrant, S., Marchant, R., Sanchez, R. M., Pandya, M., and Lopez, A. (2020). Intrinsic bias metrics do not correlate with application bias. *arXiv preprint arXiv:2012.15859*.

[Gómez-Ríos et al., 2017] Gómez-Ríos, A., Luengo, J., and Herrera, F. (2017). A study on the noise label influence in boosting algorithms: AdaBoost, GBM and XGBoost. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 268–280. Springer.

[Gretton et al., 2009] Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. (2009). Covariate shift by kernel mean matching. *Dataset Shift in Machine Learning*, 3(4):5.

[Groenwold et al., 2012] Groenwold, R. H., White, I. R., Donders, A. R. T., Carpenter, J. R., Altman, D. G., and Moons, K. G. (2012). Missing covariate data in clinical research: when and when not to use the missing-indicator method for analysis. *Canadian Medical Association Journal*, 184(11):1265–1269.

[Guo et al., 2017] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

[Gupta et al., 2016] Gupta, A., Gusain, K., and Popli, B. (2016). Verifying the value and veracity of extreme gradient boosted decision trees on a variety of datasets. In *2016 11th International Conference on Industrial and Information Systems (ICIIS)*, pages 457–462. Ieee.

[Hajian et al., 2015] Hajian, S., Domingo-Ferrer, J., Monreale, A., Pedreschi, D., and Giannotti, F. (2015). Discrimination-and privacy-aware patterns. *Data Mining and Knowledge Discovery*, 29(6):1733–1782.

[Hanley and McNeil, 1982] Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36.

[Hardt et al., 2016] Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. *arXiv preprint arXiv:1610.02413*.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). Boosting and additive trees. In *The Elements of Statistical Learning*, pages 337–387. Springer.

[Heskes, 1994] Heskes, T. (1994). *The use of being stubborn and introspective*, pages 1184–1200. Citeseer.

[Holzinger et al., 2017] Holzinger, A., Biemann, C., Pattichis, C. S., and Kell, D. B. (2017). What do we need to build explainable AI systems for the medical domain? *arXiv preprint arXiv:1712.09923*.

[Huberman and Langholz, 1999] Huberman, M. and Langholz, B. (1999). Application of the missing-indicator method in matched case-control studies with incomplete data. *American Journal of Epidemiology*, 150(12):1340–1345.

[Jacobusse and Veenman, 2016] Jacobusse, G. and Veenman, C. (2016). On selection bias with imbalanced classes. In *International Conference on Discovery Science*, pages 325–340. Springer.

[James et al., 2013] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning*, volume 112. Springer.

[Jeatrakul et al., 2010] Jeatrakul, P., Wong, K. W., and Fung, C. C. (2010). Data cleaning for classification using misclassification analysis. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 14(3):297–302.

[Jiang et al., 2020] Jiang, R., Pacchiano, A., Stepleton, T., Jiang, H., and Chiappa, S. (2020). Wasserstein fair classification. In *Uncertainty in Artificial Intelligence*, pages 862–872. PMLR.

[John, 1995] John, G. H. (1995). Robust decision trees: removing outliers from databases. In *KDD*, volume 95, pages 174–179.

[Jones, 1979] Jones, D. S. (1979). *Elementary Information Theory*. Clarendon Press.

[Kamiran et al., 2010] Kamiran, F., Calders, T., and Pechenizkiy, M. (2010). Discrimination aware decision tree learning. In *2010 IEEE International Conference on Data Mining*, pages 869–874. IEEE.

[Kamishima et al., 2012] Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. (2012). Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer.

[Kirch, 2008] Kirch, W. (2008). Pearson's correlation coefficient. *Encyclopedia of Public Health*, pages 1090–1091.

[Kleinbaum and Klein, 2010] Kleinbaum, D. G. and Klein, M. (2010). Modeling strategy guidelines. In *Logistic Regression*, pages 165–202. Springer.

[Kleinberg et al., 2016] Kleinberg, J., Mullainathan, S., and Raghavan, M. (2016). Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*.

[Knol et al., 2010] Knol, M. J., Janssen, K. J., Donders, A. R. T., Egberts, A. C., Heerdink, E. R., Grobbee, D. E., Moons, K. G., and Geerlings, M. I. (2010).

Unpredictable bias when using the missing indicator method or complete case analysis for missing confounder values: an empirical example. *Journal of Clinical Epidemiology*, 63(7):728–736.

[Kohavi et al., 1996] Kohavi, R., Wolpert, D. H., et al. (1996). Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275–83.

[Koplowitz and Brown, 1981] Koplowitz, J. and Brown, T. A. (1981). On the relation of performance to editing in nearest neighbor rules. *Pattern Recognition*, 13(3):251–255.

[Kylberg and Sintorn, 2013] Kylberg, G. and Sintorn, I.-M. (2013). Evaluation of noise robustness for local binary pattern descriptors in texture classification. *EURASIP Journal on Image and Video Processing*, 2013(1):1–20.

[Lee, 2019] Lee, J.-S. (2019). AUC4.5: AUC-based C4.5 decision tree algorithm for imbalanced data classification. *IEEE Access*, 7:106034–106042.

[Li et al., 2018] Li, R., Zhang, Y., Tuo, Y., and Chang, P. (2018). A novel method for detecting telecom fraud user. In *2018 3rd International Conference on Information Systems Engineering (ICISE)*, pages 46–50. IEEE.

[Li et al., 2015] Li, W., Mo, W., Zhang, X., Squiers, J. J., Lu, Y., Sellke, E. W., Fan, W., DiMaio, J. M., and Thatcher, J. E. (2015). Outlier detection and removal improves accuracy of machine learning approach to multispectral burn diagnostic imaging. *Journal of Biomedical Optics*, 20(12):121305.

[Lipton et al., 2016] Lipton, Z. C., Kale, D., and Wetzel, R. (2016). Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *Machine learning for healthcare conference*, pages 253–270. PMLR.

[Little, 1988] Little, R. J. (1988). A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association*, 83(404):1198–1202.

[Little and Rubin, 2019] Little, R. J. and Rubin, D. B. (2019). *Statistical Analysis with Missing Data*, volume 793. John Wiley & Sons.

[Little et al., 2014] Little, T. D., Jorgensen, T. D., Lang, K. M., and Moore, E. W. G. (2014). On the joys of missing data. *Journal of Pediatric Psychology*, 39(2):151–162.

[Liu et al., 2008] Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *2008 8th IEEE International Conference on Data Mining*, pages 413–422. IEEE.

[Liu et al., 2012] Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):1–39.

[Liu et al., 2017] Liu, H., Li, X., Li, J., and Zhang, S. (2017). Efficient outlier detection for high-dimensional data. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(12):2451–2461.

[Liu et al., 2016] Liu, J., Li, J., Li, W., and Wu, J. (2016). Rethinking big data: a review on the data quality and usage issues. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115:134–142.

[Liu and Özsu, 2009] Liu, L. and Özsu, M. T. (2009). *Encyclopedia of Database Systems*, volume 6. Springer New York, NY, USA.

[Liu and Tao, 2015] Liu, T. and Tao, D. (2015). Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461.

[Liu et al., 2014] Liu, W., Hua, G., and Smith, J. R. (2014). Unsupervised one-class learning for automatic outlier removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3826–3833.

[Malla et al., 2018] Malla, L., Perera-Salazar, R., McFadden, E., Ogero, M., Stepniewska, K., and English, M. (2018). Handling missing data in propensity score estimation in comparative effectiveness evaluations: a systematic review. *Journal of Comparative Effectiveness Research*, 7(3):271–279.

[Manwani and Sastry, 2013] Manwani, N. and Sastry, P. (2013). Noise tolerance under risk minimization. *IEEE Transactions on Cybernetics*, 43(3):1146–1151.

[Mariet et al., 2016] Mariet, Z., Harding, R., Madden, S., et al. (2016). Outlier detection in heterogeneous datasets using automatic tuple expansion. Technical report, Massachusetts Institute of Technology.

[Martínez-Plumed et al., 2019] Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Orallo, J. H., Kull, M., Lachiche, N., Quintana, M. J. R., and Flach, P. A. (2019). CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*.

[Meertens et al., 2021] Meertens, Q. A. et al. (2021). *Misclassification bias in statistical learning*. PhD thesis, Leiden University.

[Miranda et al., 2009] Miranda, A. L., Garcia, L. P. F., Carvalho, A. C., and Lorena, A. C. (2009). Use of classification algorithms in noise detection and elimination. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 417–424. Springer.

[Mitchell, 1997] Mitchell, T. (1997). *Machine Learning*. New York: McGraw Hill.

[Mok et al., 2010] Mok, M. S., Sohn, S. Y., and Ju, Y. H. (2010). Random effects logistic regression model for anomaly detection. *Expert Systems with Applications*, 37(10):7162–7166.

[Müller and Markert, 2019] Müller, N. M. and Markert, K. (2019). Identifying mislabeled instances in classification datasets. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[Mulongo et al., 2020] Mulongo, J., Atemkeng, M., Ansah-Narh, T., Rockefeller, R., Nguegnang, G. M., and Garuti, M. A. (2020). Anomaly detection in power generation plants using machine learning and neural networks. *Applied Artificial Intelligence*, 34(1):64–79.

[Muniyandi et al., 2012] Muniyandi, A. P., Rajeswari, R., and Rajaram, R. (2012). Network anomaly detection by cascading k-means clustering and C4.5 decision tree algorithm. *Procedia Engineering*, 30:174–182.

[Narkhede, 2018] Narkhede, S. (2018). Understanding AUC-ROC curve. *Towards Data Science*, 26:220–227.

[Naseer et al., 2018] Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M., and Han, K. (2018). Enhanced network anomaly detection based on deep neural networks. *IEEE Access*, 6:48231–48246.

[Niculescu-Mizil and Caruana, 2005] Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 625–632.

[Oliphant, 2007] Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science & Engineering*, 9(3):10–20.

[Olson, 2003] Olson, J. E. (2003). *Data Quality: the Accuracy Dimension*. Elsevier.

[Pafka, 2019] Pafka, S. (2019). Benchm-ml. `https://github.com/szilard/benchm-ml`.

[Painsky and Wornell, 2018] Painsky, A. and Wornell, G. (2018). On the universality of the logistic loss function. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 936–940. IEEE.

[Paris MoU, 2020] Paris MoU (2020). Current flag performance list. `https://www.parismou.org/detentions-banning/white-grey-and-black-list`.

[Park and Kwak, 2016] Park, S. and Kwak, N. (2016). Analysis on the dropout effect in convolutional neural networks. In *Asian Conference on Computer Vision*, pages 189–204. Springer.

[Pechenizkiy et al., 2006] Pechenizkiy, M., Tsymbal, A., Puuronen, S., and Pechenizkiy, O. (2006). Class noise and supervised learning in medical domains: the effect of feature extraction. In *19th IEEE Symposium on Computer-based Medical Systems (CBMS'06)*, pages 708–713. IEEE.

[Pedersen et al., 2017] Pedersen, A. B., Mikkelsen, E. M., Cronin-Fenton, D., Kristensen, N. R., Pham, T. M., Pedersen, L., and Petersen, I. (2017). Missing data and multiple imputation in clinical epidemiological research. *Clinical Epidemiology*, 9:157.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830.

[Pereira Barata, 2020] Pereira Barata, A. (2020). Crosslier detection. `https://github.com/pereirabarataap/crosslier-detection`.

[Pereira Barata, 2021] Pereira Barata, A. (2021). Fair tree classifier. `https://github.com/pereirabarataap/fair_tree_classifier`.

[Pereira Barata et al., 2018a] Pereira Barata, A., de Bruin, G. J., Takes, F. W., Veenman, C. J., and van den Herik, H. J. (2018a). Data-driven risk assessment in infrastructure networks. In *ICT.open*.

[Pereira Barata et al., 2018b] Pereira Barata, A., de Bruin, G. J., Takes, F. W., Veenman, C. J., and van den Herik, H. J. (2018b). Finding anomalies in waste transportation data with supervised category models. In *2018 27th Belgian Dutch Conference on Machine Learning (BeNeLearn)*.

[Pereira Barata et al., 2019] Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2019). Imputation methods outperform missing-indicator for data missing completely at random. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 407–414. IEEE.

[Pereira Barata et al., 2021] Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2021). The eXPose approach to crosslier detection. In

2020 25th International Conference on Pattern Recognition (ICPR), pages 2312–2319. IEEE.

[Pessach and Shmueli, 2020] Pessach, D. and Shmueli, E. (2020). Algorithmic fairness. *arXiv preprint arXiv:2001.09784*.

[Pessach and Shmueli, 2022] Pessach, D. and Shmueli, E. (2022). A review on fairness in machine learning. *ACM Computing Surveys (CSUR)*, 55(3):1–44.

[Platt et al., 1999] Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74.

[Port of Rotterdam Authority, 2021] Port of Rotterdam Authority (2021). Facts and figures. `https://www.portofrotterdam.com/sites/default/files/2021-06/facts-and-figures-port-of-rotterdam.pdf`.

[Quy et al., 2021] Quy, T. L., Roy, A., Iosifidis, V., and Ntoutsi, E. (2021). A survey on datasets for fairness-aware machine learning. *arXiv preprint arXiv:2110.00530*.

[Rausand, 2013] Rausand, M. (2013). *Risk assessment: theory, methods, and applications*, volume 115. John Wiley & Sons.

[Rekatsinas et al., 2017] Rekatsinas, T., Chu, X., Ilyas, I. F., and Ré, C. (2017). Holoclean: holistic data repairs with probabilistic inference. *arXiv preprint arXiv:1702.00820*.

[Ren et al., 2018] Ren, M., Zeng, W., Yang, B., and Urtasun, R. (2018). Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pages 4334–4343. PMLR.

[Richardson, 2022] Richardson, A. (2022). Biased data lead to biased algorithms. *CMAJ*, 194(9):E341–E341.

[Rieger et al., 2010] Rieger, A., Hothorn, T., and Strobl, C. (2010). Random forests with missing values in the covariates. Technical Report 79, University of Munich.

[Robertson, 2008] Robertson, S. (2008). A new interpretation of average precision. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 689–690.

[Rosario, 2004] Rosario, D. S. (2004). Highly effective logistic regression model for signal (anomaly) detection. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages V–817. IEEE.

[Rose and Fischer, 2011] Rose, L. T. and Fischer, K. W. (2011). Garbage in, garbage out: having useful data is everything. *Measurement: Interdisciplinary Research & Perspective*, 9(4):222–226.

[Sáez et al., 2014] Sáez, J. A., Galar, M., Luengo, J., and Herrera, F. (2014). Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition. *Knowledge and Information Systems*, 38(1):179–206.

[Saito and Rehmsmeier, 2015] Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432.

[Samuel, 1959] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229.

[Santos et al., 2019a] Santos, J. D., Chebotarov, D., McNally, K. L., Bartholomé, J., Droc, G., Billot, C., and Glaszmann, J. C. (2019a). Fine scale genomic signals of admixture and alien introgression among Asian rice landraces. *Genome Biology and Evolution*, 11(5):1358–1373.

[Santos et al., 2019b] Santos, M. S., Pereira, R. C., Costa, A. F., Soares, J. P., Santos, J., and Abreu, P. H. (2019b). Generating synthetic missing data: a review by missing mechanism. *IEEE Access*, 7:11651–11667.

[Sarker, 2021] Sarker, I. H. (2021). Machine learning: algorithms, real-world applications and research directions. *SN Computer Science*, 2(3):1–21.

[Schlomer et al., 2010] Schlomer, G. L., Bauman, S., and Card, N. A. (2010). Best practices for missing data management in counseling psychology. *Journal of Counseling Psychology*, 57(1):1.

[Schrider and Kern, 2018] Schrider, D. R. and Kern, A. D. (2018). Supervised machine learning for population genetics: a new paradigm. *Trends in Genetics*, 34(4):301–312.

[Scott, 2015] Scott, C. (2015). A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *Artificial Intelligence and Statistics*, pages 838–846. PMLR.

[Segata et al., 2010] Segata, N., Blanzieri, E., Delany, S. J., and Cunningham, P. (2010). Noise reduction for instance-based learning with a local maximal margin approach. *Journal of Intelligent Information Systems*, 35(2):301–331.

[Sharma et al., 2017] Sharma, N., Verlekar, P., Ashary, R., and Zhiquan, S. (2017). Regularization and feature selection for large dimensional data. *arXiv preprint arXiv:1712.01975*.

[Sidi et al., 2012] Sidi, F., Panahy, P. H. S., Affendey, L. S., Jabar, M. A., Ibrahim, H., and Mustapha, A. (2012). Data quality: a survey of data quality dimensions. In *2012 International Conference on Information Retrieval & Knowledge Management*, pages 300–304. IEEE.

[Stone, 1974] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133.

[Subudhi and Panigrahi, 2020] Subudhi, S. and Panigrahi, S. (2020). Use of optimized fuzzy c-means clustering and supervised classifiers for automobile insurance fraud detection. *Journal of King Saud University-Computer and Information Sciences*, 32(5):568–575.

[Sun et al., 2007] Sun, J.-w., Zhao, F.-y., Wang, C.-j., and Chen, S.-f. (2007). Identifying and correcting mislabeled training instances. In *Future Generation Communication and Networking (FGCN 2007)*, volume 1, pages 244–250. IEEE.

[Tax and Duin, 2002] Tax, D. M. and Duin, R. P. (2002). Using two-class classifiers for multiclass classification. In *Object Recognition Supported by User Interaction for Service Robots*, volume 2, pages 124–127. IEEE.

[Teng, 2000] Teng, C. M. (2000). Evaluating noise correction. In *Pacific Rim International Conference on Artificial Intelligence*, pages 188–198. Springer.

[Teng, 2001] Teng, C.-M. (2001). A comparison of noise handling techniques. In *FLAIRS Conference*, pages 269–273.

[Thongkam et al., 2008] Thongkam, J., Xu, G., Zhang, Y., and Huang, F. (2008). Support vector machine for outlier detection in breast cancer survivability prediction. In *Asia-Pacific Web Conference*, pages 99–109. Springer.

[Twala, 2009] Twala, B. (2009). An empirical comparison of techniques for handling incomplete data using decision trees. *Applied Artificial Intelligence*, 23(5):373–405.

[Van Buuren, 2018] Van Buuren, S. (2018). *Flexible Imputation of Missing Data*. CRC Press.

[Van der Heijden et al., 2006] Van der Heijden, G. J., Donders, A. R. T., Stijnen, T., and Moons, K. G. (2006). Imputation of missing values is superior to complete case analysis and the missing-indicator method in multivariable diagnostic research: a clinical example. *Journal of Clinical eEidemiology*, 59(10):1102–1109.

[Van Rijn and Hutter, 2018] Van Rijn, J. N. and Hutter, F. (2018). Hyperparameter importance across datasets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2367–2376.

[Vanschoren et al., 2014] Vanschoren, J., Van Rijn, J. N., Bischl, B., and Torgo, L. (2014). OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60.

[Vapnik, 2013] Vapnik, V. (2013). *The Nature of Statistical Learning Theory*. Springer Science & Business Media.

[Venkatasubramanian, 2019] Venkatasubramanian, S. (2019). Algorithmic fairness: measures, methods and representations. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 481–481.

[Venkatesh and Anuradha, 2019] Venkatesh, B. and Anuradha, J. (2019). A review of feature selection and its methods. *Cybernetics and Information Technologies*, 19(1):3–26.

[Vespe et al., 2012] Vespe, M., Visentini, I., Bryan, K., and Braca, P. (2012). Unsupervised learning of maritime traffic patterns for anomaly detection. In *9th IET Data Fusion & Target Tracking Conference (DF & TT 2012): Algorithms & Applications*. IET.

[Vollmer et al., 2017] Vollmer, M., Sodmann, P., Caanitz, L., Nath, N., and Kaderali, L. (2017). Can supervised learning be used to classify cardiac rhythms? In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE.

[Wang et al., 2019] Wang, X., Kodirov, E., Hua, Y., and Robertson, N. M. (2019). Derivative manipulation for general example weighting. *arXiv preprint arXiv:1905.11233*.

[Wang, 2005] Wang, Y. (2005). A multinomial logistic regression modeling approach for anomaly intrusion detection. *Computers & Security*, 24(8):662–674.

[Wilcoxon, 1992] Wilcoxon, F. (1992). Individual comparisons by ranking methods. In *Breakthroughs in Statistics*, pages 196–202. Springer.

[Wilson and Martinez, 2000] Wilson, D. R. and Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286.

[Wolpert and Macready, 1997] Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

[Woodworth et al., 2017] Woodworth, B., Gunasekar, S., Ohannessian, M. I., and Srebro, N. (2017). Learning non-discriminatory predictors. In *Conference on Learning Theory*, pages 1920–1953. PMLR.

[Wu, 1995] Wu, X. (1995). *Knowledge Acquisition from Databases*. Intellect books.

[Xia et al., 2017] Xia, Y., Liu, C., Li, Y., and Liu, N. (2017). A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, 78:225–241.

[Xiang and Min, 2010] Xiang, G. and Min, W. (2010). Applying semi-supervised cluster algorithm for anomaly detection. In *2010 Third International Symposium on Information Processing*, pages 43–45. IEEE.

[Xu et al., 2018] Xu, X., Liu, H., Li, L., and Yao, M. (2018). A comparison of outlier detection techniques for high-dimensional data. *International Journal of Computational Intelligence Systems*, 11(1):652–662.

[Yin and Dong, 2011] Yin, H. and Dong, H. (2011). The problem of noise in classification: past, current and future work. In *2011 IEEE 3rd International Conference on Communication Software and Networks*, pages 412–416. IEEE.

[Zabihi et al., 2017] Zabihi, M., Rad, A. B., Katsaggelos, A. K., Kiranyaz, S., Narkilahti, S., and Gabbouj, M. (2017). Detection of atrial fibrillation in ECG hand-held devices using a random forest classifier. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE.

[Zafar et al., 2017] Zafar, M. B., Valera, I., Rogriguez, M. G., and Gummadi, K. P. (2017). Fairness constraints: mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR.

[Zanero and Savaresi, 2004] Zanero, S. and Savaresi, S. M. (2004). Unsupervised learning techniques for an intrusion detection system. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 412–419.

[Zhang and Ntoutsi, 2019] Zhang, W. and Ntoutsi, E. (2019). FAHT: an adaptive dairness-aware decision tree classifier. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 1480–1486.

[Zhang et al., 2006] Zhang, W., Rekaya, R., and Bertrand, K. (2006). A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer. *Bioinformatics*, 22(3):317–325.

[Zhang, 2016] Zhang, Z. (2016). Missing data imputation: focusing on single imputation. *Annals of Translational Medicine*, 4(1).

[Zhu and Wu, 2004] Zhu, X. and Wu, X. (2004). Class noise vs. attribute noise: a quantitative study. *Artificial Intelligence Review*, 22(3):177–210.

Summary

In Chapter 1, we introduce the current movement towards trustworthy AI, as well as its application in risk assessment activities, which is the motivation of this work. In particular, we further elaborate on the practical use-cases within the Inspectorate of the Netherlands, in which reliable and fair models are required, given the high-risk nature of the domain. Our aim is to promote a paradigm-shift towards data-driven approaches —via machine learning techniques— to be used by the agents of the Inspectorate (i.e., inspectors).

Explicitly, the focus of this thesis is on classification models. Data generated by the Inspectorate is often tabular. As such, we focus on tree-based learning architectures. Moreover, two real-world data traits detrimental to classifier learning are considered: (1) data quality, viz missingness and noise; and (2) bias in the data. The goal is to generate, via learning techniques which combat these drawbacks, adequately-performing classifiers (measured in AUC) which are both (1) reliable, and (2) fair, respectively addressing each data trait. As such, we formulate the PS as follows.

> **PS:** *How can machine learning methods advance data-driven risk assessment by the Inspectorate in a reliable and fair manner?*

Thereafter, we decompose the PS into three tractable RQs. Below, these are stated, together with the main results gathered from answering them.

When dealing with missing data, one of three distinct mechanisms of missingness occurs: (1) MCAR; (2) MAR; or (3) MNAR (see List of Abbreviations). Depending on the missing mechanism, the efficiency of missing-data handling techniques varies. This is measured in the performance of the downstream task (i.e., classification performance) which is also dependent on the selected learner.

On the one hand, under a non-MCAR scenario, an adequate approach is to encode missingness; i.e., the missing-indicator method. On the other hand, imputation is preferred under MCAR. Although real-world data are rarely MCAR (thus justifying a missing-indicator approach), the assumption over the missing mechanism is not always true.

Although testing for MCAR is possible, the result is not a guaranteed truth. With this uncertainty in mind, the first RQ is formulated accordingly.

> **RQ1:** *Given data with missing values, which (a) missing data-handling technique and (b) learning algorithm should be jointly selected such that, regardless of the missing mechanism, the detriment to the downstream task performance is minimal when compared to the non-missing (unavailable) case?*

In Chapter 2, the MCAR scenario is studied. In a controlled environment, missing data is artificially generated for different proportions of missingness; several imputation and missing-indicator methods are deployed. Distinct classifiers are then constructed via distinct learning algorithms. The resulting performance of each learner-data handling pair is retrieved.

The results show that imputation methods provide a superior classification performance, compared to the missing-indicator method, under the MCAR scenario. Yet and most importantly, for the classifiers learned via decision tree-based gradient boosting, the differences in performance derived from the two distinct data-handling techniques becomes negligible. Hence, the answer to RQ1 is that the missing-indicator method, in conjunction with a decision tree-based learner —particularly via gradient boosting— should be used regardless of the missing mechanism.

Noise in data deteriorates the classification performance and may present itself as either feature noise or (class) label noise, of which the latter is more detrimental. Under label noise, three noise-generating mechanisms exist: (1) NCAR; (2) NAR; and (3) NNAR. Handling noise in data generally entails generating noisy-sample detection scores, traditionally addressed by leveraging classifiers learned on the noisy data, which is an endeavour in itself.

Within the scenario of the Inspectorate, noisy data may represent misconduct; e.g., companies manipulating waste transportation reports to lower the costs associated with each waste type. Therefore, it is of importance to detect these misconducts and simultaneously learn classifiers from the available data which contain them. Hence, the second RQ is a compound one and decomposed into RQ2(a) and RQ2(b).

> **RQ2:** *Given data with label noise, how can noisy-samples be (a) adequately detected, and (b) used to learn a well-performing model?*

In Chapter 3, we introduce the term *crosslier*. It denotes a sample with disharmonious feature values. Concretely, crossliers are a special case of outlier with respect to some overarching category feature. They are samples which exhibit label noise with respect to the category feature, and potentially feature noise, relating to the NNAR mechanism. To detect crossliers, we propose the EXPOSE method.

The EXPOSE method evaluates samples in a CV-manner, such that all samples in the data are evaluated. Each training set is used to produce a well-calibrated classifier via Platt scaling. The classifier is then deployed on the corresponding test samples. From the classifier output $f(x)$, the crosslier score is $-\log_2[f(x)]$. By evaluating the performance of our method in a controlled setup, we validate EXPOSE. Then, we answer RQ2(a).

Chapter 4 follows logically, utilising the core principal of EXPOSE —and its established validity— towards classifier learning with noisy data. We term our compound method DENOISE. The DENOISE method entails two steps.

First, well-calibrated probabilities are computed for each sample following the EXPOSE method. Second, the probabilities are used to generate individual sample weights, such that the weight is the log-odds of the output sample probability. Under a logistic loss function applied via a gradient boosting decision tree learner, an adequately-performing noise-resilient classifier is produced. In a controlled experimental environment, we validate our method, thereby answering RQ2(b).

Learning from biased data leads to biased models. Even when the conditions of the data gathering processes are ideal, bias in data may still occur due to historical factors (e.g., gender wage gap). To address this issue, different fair machine learning techniques can be used. The purpose of these techniques is to generate models of which the output is independent of some sensitive attribute, such as gender; i.e., fair models. Moreover, several measures of fairness exist, of which the strong demographic parity is analogous to the AUC performance measure. Generally, the greater the fairness of a model, the lesser its predictive performance: here we speak of the performance-fairness trade-off.

Regarding the Inspectorate, biased data may represent a form of bias given some distinct criterion. For example, ships sailing under specific country flags are deemed of higher risk and hence more targetable than others, which may result in confirmation bias. Moreover, since the country flag is easily mutable, it enables companies to bypass the inspection selection protocol. From these considerations, we formulate our RQ3.

> **RQ3:** *How can we, from biased data, learn a model tunable with respect to the performance-fairness trade-off such that the selection of the trade-off point is made intuitive for the relevant stake-holders?*

In Chapter 5, we propose a fair decision tree learning algorithm via strong demographic parity. We do so by defining a compound splitting criterion, termed SCAFF —splitting criterion AUC for fairness— which is tunable with respect to the performance-fairness trade-off. It leverages several sensitive attributes concurrently, of which the values may be multicategorical.

SCAFF is defined as a weighted linear combination of (a) the traditional AUC classification performance, and (b) the strong demographic parity scaled to the range of the AUC. We term the scaled fairness measure *sensitive* AUC. The closer the sensitive AUC is to 0.5, the greater the fairness of the model. By incorporating an orthogonality parameter $\Theta \in [0, 1]$ implemented as an elastic net-like weight to the performance and the fairness terms, the performance-fairness trade-off is tunable. In the case of $\Theta = 0$, a traditional (potential) non-fair classifier is generated, and increasing Θ augments the fairness of the final model. By comparing SCAFF to other fair splitting criteria in a controlled experiment, we validate our approach and answer RQ3.

The conclusion of the thesis is that, under the current movement towards trustworthy AI in Europe, shifting the current risk assessment paradigm to a more data-driven methodology is a delicate yet feasible venture via reliable and fair machine learning. Given the high-risk nature of risk assessment activities, and the characteristics of the data generated by them, we believe that technical methods can ensure the adequacy of the final learned models. In particular, the issues associated with learning a classification model from biased and low-quality data can be successfully addressed, producing adequate models.

Samenvatting

Het inschatten van risico's is het hoofdonderwerp van dit proefschrift. In Hoofdstuk 1 introduceren we in algemene zin de recente ontwikkelingen op het gebied van kunstmatige intelligentie (Artificial Intelligence, AI). We kijken daarbij vooral naar het gebruik van AI-technieken bij het inschatten van risico's. In het bijzonder besteden we aandacht aan de praktische inzetbaarheid van AI-programma's bij de Inspectie Leefomgeving en Transport (ILT). Binnen dit orgaan van de Nederlandse overheid is grote behoefte aan een *eerlijke* en *betrouwbare* inschatting van risico's. Het onderzoekswerk is een bijdrage aan de paradigma-verschuiving die in 2016 in gang gezet is bij de start van het programma Anders Omgaan Met Data (AOMD). Ons werk draagt dan ook bij aan de voortdurende verbetering van het data-gestuurd werken door de ILT-inspecteurs, met name door middel van de inzet van *machine learning*.

De focus van dit proefschrift ligt expliciet op classificatiemodellen. Data die door de inspectie wordt gegenereerd, verzameld en geclassificeerd staan vaak (a) in tabelvorm en (b) bevatten diverse datatypen. In het onderzoekswerk richten we ons met name op data in beslisbomen. Daarbij onderzoeken we twee problemen die regelmatig voorkomen bij classificatie: (1) *bias in de data*; en (2) *datakwaliteit*. Bij datakwaliteit kijken we in het bijzonder naar welke data er missen (*missing data*) en ruis (*noise*). Het doel is om adequaat presterende classificatiemodellen te ontwerpen die deze problemen oplossen, en dus zowel *eerlijk* als *betrouwbaar* zijn. De probleemstelling van dit proefschrift luidt dan ook als volgt.

> **Probleemstelling:** *Hoe kunnen we bijdragen aan eerlijke en betrouwbare machine learning methoden, die inzetbaar zijn voor de inschatting van risico's door de Inspectie Leefomgeving en Transport (ILT)?*

De probleemstelling wordt onderverdeeld in drie onderzoeksvragen. Hieronder bespreken we deze vragen, geven dan de precieze formulering en vatten tenslotte de belangrijkste bevindingen samen.

Bij *missing data* speelt een precieze karakterisering van het mechanisme (waardoor ontbreken de data eigenlijk?) een belangrijke rol. We onderscheiden drie mechanismen: (1) MCAR (Missing Completely At Random); (2) MAR (Missing At Random); en (3) MNAR (Missing Not At Random). Afhankelijk van het mechanisme waar we mee te maken hebben, zien we dat de efficiency van de *missing-data* technieken nogal varieert. In ons onderzoek meten we de performance van de verschillende technieken en kijken we naar de relatie tussen de performance en het gekozen *machine learning* model.

In een *niet-MCAR*-scenario (dus MAR en MNAR) lijkt een adequate benadering te zijn: het coderen van ontbrekende gegevens door een *extra variabele* (of attribuut). Dit is de zogeheten *missing-indicator* methode. Bij het *MCAR*-scenario heeft daarentegen —zo lijkt het— *imputatie* van de *missing data* de voorkeur. In de werkelijke wereld ontbreekt data echter zelden volgens het MCAR mechanisme; bovendien is het testen of data ontbreekt volgens het MCAR mechanisme moeilijk en soms ook nog onbetrouwbaar. Daarom luidt de eerste onderzoeksvraag als volgt.

> **Onderzoeksvraag 1:** *Welke combinatie van methoden voor het behandelen van missing data en machine learning algoritmen moet worden gebruikt om, los van het precieze missing data mechanisme, een adequaat presterend model te verkrijgen?*

In Hoofdstuk 2 wordt het MCAR mechanisme bestudeerd. Door op een gecontroleerde manier verschillende gradaties van *missing data* volgens het MCAR mechanisme te genereren, kan empirisch worden vergeleken hoe combinaties van bepaalde methoden voor het behandelen van *missing data* en *machine learning* algoritmen presteren. De resultaten laten zien dat onder het MCAR mechanisme *imputatiemethoden doorgaans beter presteren dan de missing-indicator methode.*

Een belangrijke bevinding is dat voor *gradient boosting* classificatie-algoritmen op basis van beslisbomen, de verschillen in prestaties verwaarloosbaar lijken. Het antwoord op de eerste onderzoeksvraag is dan ook dat de missing-indicator methode, in combinatie met een beslisboomalgoritme — met name op basis van *gradient boosting* — moet worden gebruikt, ongeacht het precieze mechanisme dat aangeeft welke data er ontbreekt.

Door ruis in de data zullen in het algemeen de classificatie-prestaties verslechteren. Ruis kan voorkomen als *featureruis* of als (klasse) *labelruis*. Vooral labelruis heeft een sterke invloed. Bij labelruis bestaan drie mechanismen: (1) NCAR (Noise Completely At Random); (2) NAR (Noise At Random); en (3) NNAR (Noise Not At Random). De aanpak bij ruis in de data richt zich in het algemeen op het bepalen van detectiescores voor de ruisdata.

De detectiescores worden traditioneel bepaald door gebruik te maken van classificatiescores die zijn geleerd op de ruis bevattende dataset. Binnen het scenario van de Inspectie kunnen data met labelruis juist een aanwijzing zijn voor afwijkingen, vooral wanneer er ook featureruis is (d.w.z., volgens NNAR mechanisme); bijv., bedrijven die afvaltransport-rapporten manipuleren om de kosten ervan te verlagen. Het is van belang om niet alleen deze afwijkingen op te sporen, maar ook om zo goed mogelijk classificatiemodellen te leren uit de beschikbare gegevens.

Binnen het scenario van de Inspectie kunnen data met labelruis juist een aanwijzing zijn voor afwijkingen, vooral wanneer er ook featureruis is (d.w.z., volgens NNAR mechanisme); b.v. bedrijven die afvaltransportrapporten manipuleren om de kosten ervan te verlagen. Het is van belang om niet alleen deze afwijkingen op te sporen, maar ook om zo goed mogelijk classificatiemodellen te leren uit de beschikbare gegevens.

De tweede onderzoeksvraag is samengesteld uit onderzoeksvraag 2(a) en onderzoeksvraag 2(b) en luidt in zijn geheel als volgt.

> **Onderzoeksvraag 2:** *Gegeven data met labelruis, hoe kan die data met ruis (a) adequaat worden opgespoord, en (b) worden gebruikt om een model met adequate performance te leren?*

In Hoofdstuk 3 introduceren we de term *crosslier*, die een sample aanduidt met afwijkende features. Preciezer gezegd crossliers zijn een speciaal type outlier die afwijkend zijn ten opzichte van een bepaalde categorie. Het zijn *samples* die labelruis vertonen met betrekking tot een bepaalde categorische feature, waarbij de ruis mogelijk het NNAR-mechanisme volgt. Om crossliers te detecteren, stellen we de onderzoeksmethode EXPOSE voor. De EXPOSE-methode evalueert *samples* op een cross-validatie (CV) manier, zodat uiteindelijk alle *samples* worden geëvalueerd. In de *CV loops* wordt elke training set gebruikt om een goed gecalibreerde *classifier* te produceren daarbij gebruik makend van Platt-schaling. De *classifier* wordt vervolgens ingezet op de bijbehorende *test samples*. Van de *classificatie output* $f(x)$, is de corresponderende crosslierscore $-\log_2[f(x)]$. Om de prestaties van onze aanpak te evalueren, valideren we de EXPOSE-methode in een gecontroleerde experimentele omgeving. Daarmee beantwoorden we onderzoeksvraag 2(a).

Hoofdstuk 4 bouwt voort op de uitgangspunten van EXPOSE en levert een aanpak op voor het leren van een classificatiemodel met gegevens die ruis bevatten. We noemen deze samengestelde methode DENOISE. De DENOISE methode bestaat uit twee stappen. Eerst worden goed gecalibreerde kansen berekend voor elk *sample* volgens de EXPOSE-methode. Ten tweede worden de kansen gebruikt om individuele *sample* gewichten te genereren als de *log-odds* van de gecalibreerde *sample* kans.

Tezamen met een logistische verliesfunctie die wordt toegepast op de leeralgoritme van de *gradient boosting* beslissingsboom ontstaat een adequaat presterende ruisbestendige *classifier*. In een gecontroleerde experimentele omgeving valideren we vervolgens onze methode. Daarmee beantwoorden we onderzoeksvraag 2(b).

Leren van gegevens met een bias leidt tot modellen met een bias. Zelfs wanneer de omstandigheden van het verzamelen van gegevens ideaal zijn, kan er nog steeds vertekening in de gegevens optreden als gevolg van historische factoren (bijv., de loonkloof tussen mannen en vrouwen). Om dit probleem aan te pakken, kunnen verschillende eerlijke (*fair*) *machine learning*-technieken worden gebruikt. Het doel van deze technieken is om modellen te genereren waarvan de output onafhankelijk is van een gevoelig attribuut, zoals geslacht; dat wil zeggen, eerlijke (*fair*) modellen. Let wel, er bestaan verschillende maatstaven voor eerlijkheid, waarvan *strong demographic parity* analoog is aan de AUC prestatiemaatstaf voor classificatie *performance*. Over het algemeen geldt dat *fairness* op gespannen voet staat met de voorspellende kracht van *performance*. Het betreft hier de *trade-off* tussen *performance* en *fairness* gegeven een bepaald criterium.

Een voorbeeld is als schepen onder de vlag van een bepaald land varen, dan kunnen ze worden beschouwd als schepen met een hoger risico. Dit kan betekenen dat ze daarom meer aandacht krijgen dan andere schepen, wat kan leiden tot *confirmation bias*. Aangezien de landsvlag gemakkelijk kan worden gewijzigd, kunnen bedrijven het inspectie-selectieprotocol betrekkelijk gemakkelijk omzeilen. Op grond van deze observaties (en gevolgtrekkingen) formuleren we de onderzoeksvraag 3 als volgt.

> **Onderzoeksvraag 3:** *Hoe kunnen we een model bouwen van vooringenomen gegevens, zodat het door de domeinexpert kan worden aangepast met betrekking tot de trade-off tussen performance en fairness?*

In Hoofdstuk 5, we stellen een eerlijk algoritme voor om een beslisboom te leren voor *strong demographic parity*. We doen dit door een samengesteld splitsingscriterium te definiëren, genaamd SCAFF (Splitsings Criterium AUC For Fairness). SCAFF kan worden *getuned* met betrekking tot de *trade-off* tussen *performance* en *fairness*. Tegelijkertijd kan het mechanisme gebruik maken van verschillende gevoelige attributen, waarvan de waarden multi-categorisch kunnen zijn.

SCAFF wordt derhalve gedefinieerd als een gewogen lineaire combinatie van (a) de traditionele AUC classificatieperformance , en (b) de *strong demographic parity* die geschaald is in overeenstemming met het bereik van de AUC. We noemen deze AUC een *sensitive AUC*. Hoe dichter de sensitive AUC bij 0.5 ligt, des te eerlijker is het model.

Door een orthogonaliteitsparameter $\Theta \in [0,1]$ op te nemen, die geïmplementeerd wordt als een *elastisch net*-achtig gewicht op de *trade-off* tussen *performance* en *fairness*, kan de waarde worden aangepast; $\Theta = 0$ genereert een traditioneel (potentieel) niet-eerlijke classificatie, en het verhogen van Θ vergroot de *fairness* van het uiteindelijke model. Door SCAFF te vergelijken met andere eerlijke splitsingscriteria in een gecontroleerd experiment, valideren we onze aanpak en beantwoorden we de derde onderzoeksvraag.

De *conclusie* van het proefschrift wordt gegeven in Hoofdstuk 6. Door de nadruk op betrouwbare AI in Europa zien we een verschuiving van het huidige paradigma van risicobeoordeling naar een meer data-gestuurde beoordeling. Het resultaat is een delicate maar haalbare onderneming via eerlijke en betrouwbare *machine learning*. Met inachtneming van het risicovolle karakter van de beoordelingsactiviteiten en de kenmerken van de data die door de beoordelingen worden gegenereerd, kunnen technische methoden de geschiktheid van de uiteindelijk geleerde modellen garanderen. Met name de problemen die samenhangen met het leren van een classificatiemodel uit data met een bias (en met lage kwaliteit) kunnen hiermee worden aangepakt.

Curriculum Vitae

António Pedro Pereira Barata was born in Lisbon, Portugal, on the 21st of December 1989. He completed his B.Sc. in Biology in 2013, specialising in Evolutionary and Developmental Biology, in the Faculty of Science of the University of Lisbon. Within the same institute, he acquired his M.Sc. in Bioinformatics and Computational Biology in 2017. Thereafter, he was admitted to Leiden University, the Netherlands, to conduct the research presented in this doctoral thesis, under the attentive care and supervision of H. Jaap van den Herik, Cor J. Veenman, and Frank W. Takes. At the time of writing, he is employed by the Ministry of Infrastructure and Water Management of the Netherlands, within the Innovation and Data Lab of the ILT.

Publications

While working towards this thesis, the following contributions were made.

- Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2022). Fair tree classifier using strong demographic parity. *Machine Learning (under review)*
- Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2022). Noise-resilient classifier learning. *Pattern Recognition (under review)*
- Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2021). The eXPose approach to crosslier detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2312–2319. IEEE
- Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2019). Imputation methods outperform missing-indicator for data missing completely at random. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 407–414. IEEE
- Pereira Barata, A., de Bruin, G. J., Takes, F. W., Veenman, C. J., and van den Herik, H. J. (2018b). Finding anomalies in waste transportation data with supervised category models. In *2018 27th Belgian Dutch Conference on Machine Learning (BeNeLearn)*
- Pereira Barata, A., de Bruin, G. J., Takes, F. W., Veenman, C. J., and van den Herik, H. J. (2018a). Data-driven risk assessment in infrastructure networks. In *ICT.open*
- de Bruin, G. J., Pereira Barata, A., van den Herik, H. J., Takes, F. W., and Veenman, C. J. (2022). Fair automated assessment of noncompliance in cargo ship networks. *EPJ Data Science*, 11(1):13
- Angenent, M. N., Pereira Barata, A., and Takes, F. W. (2020). Large-scale machine learning for business sector prediction. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 1143–1146

Acknowledgements

A thesis is not something one can muster alone. Throughout this long and arduous journey, I had the privilege of meeting, working with, and learning from some of the most fascinating people I had ever come across.

I would like to thank Cor Veenman, for all his wisdom and patience in teaching me how to do research in machine learning. Many a *fiery* discussion were had between the two of us. I sincerely hope you enjoyed them as much as I did. In extension, I thank the Netherlands Organisation for Applied Scientific Research (TNO).

I thank Frank Takes for his forever good disposition, and for teaching me that I too could be a *glass half-full* kind of person; not only towards myself, but especially towards others. I honestly believe you have made me a better person. In extension, I thank the Leiden Institute of Advanced Computer Science (LIACS).

To Jaap van den Herik, I will forever be grateful: thank you for promoting the best in me, for teaching me to pursue the *exceptional* rather than the *passable*. This thesis would not be as it is without your aid. In extension, I thank the Leiden Centre of Data Science (LCDS).

I would like to thank also Gerrit-Jan de Bruin, for being not only my colleague but also my friend. You were the only person who *really* understood what this project entailed, its ups and downs. We have been on this same PhD boat for years now, and I eagerly look forward to our next trip. In your own boat. You own a boat. Please invite me to your boat.

I would like to express my thanks to the Innovation and Data Lab team. Particularly, I thank Jasper van Vliet for doing his utmost to integrate me in the team and the challenges we tackled, Arjan van der Put for our many adventures in the land of trial-and-error, Corline Koolhaas for her continuous concern for my well-being especially during the last stages of the thesis, and Tony Liebregts for (amongst other things) our delightful —yet short— afternoon in Lisbon. I extend my thanks also to the ILT of the Ministry of Infrastructure and Water Management, especially thanking Jan van den Bos without whom none of this work would have been possible.

With regards to Alexa and Patins, I have no idea how you two were able to put up with me during these PhD years. Without you both, I would have crumbled, and this thesis would not exist. I could not have asked for better people with whom to share my daily struggles. You were (almost) always supportive, (almost) always understanding, (almost) always impeccable; but most importantly, you were *always* there for me.

To all my family, to all my friends, to all the unnamed heroes who shaped me into what I am today: thank you.

SIKS Dissertation Series

2016 01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines

02 Michiel Christiaan Meulendijk (UU), Optimizing Medication Reviews Through Decision Support: Prescribing a Better Pill to Swallow

03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support

04 Laurens Rietveld (VU), Publishing and Consuming Linked Data

05 Evgeny Sherkhonov (UVA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers

06 Michel Wilson (TUD), Robust Scheduling in an Uncertain Environment

07 Jeroen de Man (VU), Measuring and Modeling Negative Emotions for Virtual Training

08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks From Unstructured Data

09 Archana Nottamkandath (VU), Trusting Crowdsourced Information on Cultural Artefacts

10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms

11 Anne Schuth (UVA), Search Engines That Learn From Their Users

12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems

13 Nana Baah Gyan (VU), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach

14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization

15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments

16 Guangliang Li (UVA), Socially Intelligent Autonomous Agents That Learn From Human Reward

17 Berend Weel (VU), Towards Embodied Evolution of Robot Organisms

18 Albert Meroño Peñuela (VU), Refining Statistical Data on the Web

19 Julia Efremova (Tu/e), Mining Social Structures From Genealogical Data

20 Daan Odijk (UVA), Context & Semantics in News & Web Search

21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground

22 Grace Lewis (VU), Software Architecture Strategies for Cyber-Foraging Systems

23 Fei Cai (UVA), Query Auto Completion in Information Retrieval

24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and Data Model Independent Approach

25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior

26 Dilhan Thilakarathne (VU), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, With Applications in Aviation and Energy Management Domains

27 Wen Li (TUD), Understanding Geo-Spatial Information on Social Media

28 Mingxin Zhang (TUD), Large-Scale Agent-Based Social Simulation - A Study on Epidemic Prediction and Control

29 Nicolas Höning (TUD), Peak Reduction in Decentralised Electricity Systems - Markets and Prices for Flexible Planning

30 Ruud Mattheij (UvT), The Eyes Have It

31 Mohammad Khelghati (UT), Deep Web Content Monitoring

32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations

33 Peter Bloem (UVA), Single Sample Statistics, Exercises in Learning From Just One Example

34 Dennis Schunselaar (TUE), Configurable Process Trees: Elicitation, Analysis, and Enactment

35 Zhaochun Ren (UVA), Monitoring Social Media: Summarization, Classification and Recommendation

36 Daphne Karreman (UT), Beyond R2D2: The Design of Nonverbal Interaction Behavior Optimized for Robot-Specific Morphologies

37 Giovanni Sileno (UvA), Aligning Law and Action - A Conceptual and Computational Inquiry

38 Andrea Minuto (UT), Materials That Matter - Smart Materials Meet Art & Interaction Design

39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect

40 Christian Detweiler (TUD), Accounting for Values in Design

41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance

42 Spyros Martzoukos (UVA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora

43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice

44 Thibault Sellam (UVA), Automatic Assistants for Database Exploration

45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control

46 Jorge Gallego Perez (UT), Robots to Make You Happy

47 Christina Weber (UL), Real-Time Foresight - Preparedness for Dynamic Innovation Networks

48 Tanja Buttler (TUD), Collecting Lessons Learned

49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis

50 Yan Wang (UVT), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

---

2017 01 Jan-Jaap Oerlemans (UL), Investigating Cybercrime

02 Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks Using Argumentation

03 Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach With Autonomous Products and Reconfigurable Manufacturing Machines

04 Mrunal Gawade (CWI), Multi-Core Parallelism in a Column-Store

05 Mahdieh Shadi (UVA), Collaboration Behavior

06 Damir Vandic (EUR), Intelligent Information Systems for Web Product Search

07 Roel Bertens (UU), Insight in Information: From Abstract to Anomaly

08 Rob Konijn (VU), Detecting Interesting Differences:Data Mining in Health Insurance Data Using Outlier Detection and Subgroup Discovery

09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text

10 Robby van Delden (UT), (Steering) Interactive Play Behavior

11 Florian Kunneman (RUN), Modelling Patterns of Time and Emotion in Twitter #Anticipointment

12 Sander Leemans (TUE), Robust Process Mining With Guarantees

13 Gijs Huisman (UT), Social Touch Technology - Extending the Reach of Social Touch Through Haptic Technology

14 Shoshannah Tekofsky (UvT), You Are Who You Play You Are: Modelling Player Traits From Video Game Behavior

15 Peter Berck (RUN), Memory-Based Text Correction

16 Aleksandr Chuklin (UVA), Understanding and Modeling Users of Modern Search Engines

17 Daniel Dimov (UL), Crowdsourced Online Dispute Resolution

18 Ridho Reinanda (UVA), Entity Associations for Search

19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval

20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility

21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)

22 Sara Magliacane (VU), Logics for Causal Inference Under Uncertainty

23 David Graus (UVA), Entities of Interest — Discovery in Digital Traces

24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning

25 Veruska Zamborlini (VU), Knowledge Representation for Clinical Guidelines, With Applications to Multimorbidity Analysis and Literature Search

26 Merel Jung (UT), Socially Intelligent Robots That Understand and Respond to Human Touch

27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors

28 John Klein (VU), Architecture Practices for Complex Contexts

29 Adel Alhuraibi (UvT), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"

30 Wilma Latuny (UvT), The Power of Facial Expressions

31 Ben Ruijl (UL), Advances in Computational Methods for QFT Calculations

32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives

33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity

34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics

35 Martine de Vos (VU), Interpreting Natural Science Spreadsheets

36 Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation From High-Throughput Imaging

37 Alejandro Montes Garcia (TUE), WiBAF: A Within Browser Adaptation Framework That Enables Control Over Privacy

38 Alex Kayal (TUD), Normative Social Applications

39 Sara Ahmadi (RUN), Exploiting Properties of the Human Auditory System and Compressive Sensing Methods to Increase Noise Robustness in ASR

40 Altaf Hussain Abro (VUA), Steer Your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support for Applications in Human-Aware Support Systems

41 Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle

42 Elena Sokolova (RUN), Causal Discovery From Mixed and Missing Data With Applications on ADHD Datasets

43 Maaike de Boer (RUN), Semantic Mapping in Video Retrieval

44 Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering

45 Bas Testerink (UU), Decentralized Runtime Norm Enforcement

46 Jan Schneider (OU), Sensor-Based Learning Support

47 Jie Yang (TUD), Crowd Knowledge Creation Acceleration

48 Angel Suarez (OU), Collaborative Inquiry-Based Learning

2018 01 Han van der Aa (VUA), Comparing and Aligning Process Representations

02 Felix Mannhardt (TUE), Multi-Perspective Process Mining

03 Steven Bosems (UT), Causal Models for Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction

04 Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks

05 Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Information Seeking Process

06 Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems

07 Jieting Luo (UU), A Formal Account of Opportunism in Multi-Agent Systems

08 Rick Smetsers (RUN), Advances in Model Learning for Software Systems

09 Xu Xie (TUD), Data Assimilation in Discrete Event Simulations

10 Julienka Mollee (VUA), Moving Forward: Supporting Physical Activity Behavior Change Through Intelligent Technology

11 Mahdi Sargolzaei (UVA), Enabling Framework for Service-Oriented Collaborative Networks

12 Xixi Lu (TUE), Using Behavioral Context in Process Mining

13 Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future

14 Bart Joosten (UVT), Detecting Social Signals With Spatiotemporal Gabor Filters

15 Naser Davarzani (UM), Biomarker Discovery in Heart Failure

16 Jaebok Kim (UT), Automatic Recognition of Engagement and Emotion in a Group of Children

17 Jianpeng Zhang (TUE), On Graph Sample Clustering

18 Henriette Nakad (UL), De Notaris en Private Rechtspraak

19 Minh Duc Pham (VUA), Emergent Relational Schemas for RDF

20 Manxia Liu (RUN), Time and Bayesian Networks

21 Aad Slootmaker (OUN), EMERGO: A Generic Platform for Authoring and Playing Scenario-Based Serious Games

22 Eric Fernandes de Mello Araujo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks

23 Kim Schouten (EUR), Semantics-Driven Aspect-Based Sentiment Analysis

24 Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots

25 Riste Gligorov (VUA), Serious Games in Audio-Visual Collections

26 Roelof Anne Jelle de Vries (UT), Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology

27 Maikel Leemans (TUE), Hierarchical Process Mining for Scalable Software Analysis

28 Christian Willemse (UT), Social Touch Technologies: How They Feel and How They Make You Feel

29 Yu Gu (UVT), Emotion Recognition From Mandarin Speech

30 Wouter Beek (VUA), The ”K” in ”Semantic Web” Stands for ”Knowledge”: Scaling Semantics to the Web

---

2019 01 Rob van Eijk (UL), Web Privacy Measurement in Real-Time Bidding Systems. A Graph-Based Approach to RTB System Classification

02 Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty

03 Eduardo Gonzalez Lopez de Murillas (TUE), Process Mining on Databases: Extracting Event Data From Real Life Data Sources

04 Ridho Rahmadi (RUN), Finding Stable Causal Structures From Clinical Data

05 Sebastiaan van Zelst (TUE), Process Mining With Streaming Data

06 Chris Dijkshoorn (VU), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets

07 Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms

08 Frits de Nijs (TUD), Resource-Constrained Multi-Agent Markov Decision Processes

09 Fahimeh Alizadeh Moghaddam (UVA), Self-Adaptation for Energy Efficiency in Software Systems

10 Qing Chuan Ye (EUR), Multi-Objective Optimization Methods for Allocation and Prediction

11 Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs

12 Jacqueline Heinerman (VU), Better Together

13 Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation

14 Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses

15 Erwin Walraven (TUD), Planning Under Uncertainty in Constrained and Partially Observable Environments

16 Guangming Li (TUE), Process Mining Based on Object-Centric Behavioral Constraint (OCBC) Models

17 Ali Hurriyetoglu (RUN), Extracting Actionable Information From Microtexts

18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication

19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents

20 Chide Groenouwe (UU), Fostering Technically Augmented Human Collective Intelligence

21 Cong Liu (TUE), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection

22 Martin van den Berg (VU), Improving IT Decisions With Enterprise Architecture

23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification

24 Anca Dumitrache (VU), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing

25 Emiel van Miltenburg (VU), Pragmatic Factors in (Automatic) Image Description

26 Prince Singh (UT), An Integration Platform for Synchromodal Transport

27 Alessandra Antonaci (OUN), The Gamification Design Process Applied to (Massive) Open Online Courses

28 Esther Kuindersma (UL), Cleared for Take-Off: Game-Based Learning to Prepare Airline Pilots for Critical Situations

29 Daniel Formolo (VU), Using Virtual Agents for Simulation and Training of Social Skills in Safety-Critical Circumstances

30 Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems

31 Milan Jelisavcic (VU), Alive and Kicking: Baby Steps in Robotics

32 Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games

33 Anil Yaman (TUE), Evolution of Biologically Inspired Learning in Artificial Neural Networks

34 Negar Ahmadi (TUE), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES

35 Lisa Facey-Shaw (OUN), Gamification With Digital Badges in Learning Programming

36 Kevin Ackermans (OUN), Designing Video-Enhanced Rubrics to Master Complex Skills

37 Jian Fang (TUD), Database Acceleration on FPGAs

38 Akos Kadar (OUN), Learning Visually Grounded and Multilingual Representations

---

2020 01 Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour

02 Marcos de Paula Bueno (UL), Unraveling Temporal Processes Using Probabilistic Graphical Models

03 Mostafa Deghani (UvA), Learning With Imperfect Supervision for Language Understanding

04 Maarten van Gompel (RUN), Context as Linguistic Bridges

05 Yulong Pei (TUE), On Local and Global Structure Mining

06 Preethu Rose Anish (UT), Stimulation Architectural Thinking During Requirements Elicitation - An Approach and Tool Support

07 Wim van der Vegt (OUN), Towards a Software Architecture for Reusable Game Components

08 Ali Mirsoleimani (UL), Structured Parallel Programming for Monte Carlo Tree Search

09 Myriam Traub (UU), Measuring Tool Bias and Improving Data Quality for Digital Humanities Research

10 Alifah Syamsiyah (TUE), In-Database Preprocessing for Process Mining

11 Sepideh Mesbah (TUD), Semantic-Enhanced Training Data AugmentationMethods for Long-Tail Entity Recognition Models

12 Ward van Breda (VU), Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment

13 Marco Virgolin (CWI), Design and Application of Gene-Pool Optimal Mixing Evolutionary Algorithms for Genetic Programming

14 Mark Raasveldt (CWI/UL), Integrating Analytics With Relational Databases

15 Konstantinos Georgiadis (OUN), Smart CAT: Machine Learning for Configurable Assessments in Serious Games

16 Ilona Wilmont (RUN), Cognitive Aspects of Conceptual Modelling

17 Daniele Di Mitri (OUN), The Multimodal Tutor: Adaptive Feedback From Multimodal Experiences

18 Georgios Methenitis (TUD), Agent Interactions & Mechanisms in Markets With Uncertainties: Electricity Markets in Renewable Energy Systems

19 Guido van Capelleveen (UT), Industrial Symbiosis Recommender Systems

20 Albert Hankel (VU), Embedding Green ICT Maturity in Organisations

21 Karine da Silva Miras de Araujo (VU), Where Is the Robot?: Life as It Could Be

22 Maryam Masoud Khamis (RUN), Understanding Complex Systems Implementation Through a Modeling Approach: The Case of E-Government in Zanzibar

23 Rianne Conijn (UT), The Keys to Writing: A Writing Analytics Approach to Studying Writing Processes Using Keystroke Logging

24 Lenin da Nobrega Medeiros (VUA/RUN), How Are You Feeling, Human? Towards Emotionally Supportive Chatbots

25 Xin Du (TUE), The Uncertainty in Exceptional Model Mining

26 Krzysztof Leszek Sadowski (UU), GAMBIT: Genetic Algorithm for Model-Based Mixed-Integer opTimization

27 Ekaterina Muravyeva (TUD), Personal Data and Informed Consent in an Educational Context

28 Bibeg Limbu (TUD), Multimodal Interaction for Deliberate Practice: Training Complex Skills With Augmented Reality

29 Ioan Gabriel Bucur (RUN), Being Bayesian About Causal Inference

30 Bob Zadok Blok (UL), Creatief, Creatieve, Creatiefst

31 Gongjin Lan (VU), Learning Better – From Baby to Better

32 Jason Rhuggenaath (TUE), Revenue Management in Online Markets: Pricing and Online Advertising

33 Rick Gilsing (TUE), Supporting Service-Dominant Business Model Evaluation in the Context of Business Model Innovation

34 Anna Bon (MU), Intervention or Collaboration? Redesigning Information and Communication Technologies for Development

35 Siamak Farshidi (UU), Multi-Criteria Decision-Making in Software Production

---

2021 01 Francisco Xavier Dos Santos Fonseca (TUD), Location-Based Games for Social Interaction in Public Space

02 Rijk Mercuur (TUD), Simulating Human Routines:Integrating Social Practice Theory in Agent-Based Models

03 Seyyed Hadi Hashemi (UVA), Modeling Users Interacting With Smart Devices

04 Ioana Jivet (OU), The Dashboard That Loved Me: Designing Adaptive Learning Analytics for Self-Regulated Learning

05 Davide Dell'Anna (UU), Data-Driven Supervision of Autonomous Systems

06 Daniel Davison (UT), "Hey Robot, What Do You Think?" How Children Learn With a Social Robot

07 Armel Lefebvre (UU), Research Data Management for Open Science

08 Nardie Fanchamps (OU), The Influence of Sense-Reason-Act Programming on Computational Thinking

09 Cristina Zaga (UT), The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Children's Collaboration Through Play

10 Quinten Meertens (UvA), Misclassification Bias in Statistical Learning

11 Anne van Rossum (UL), Nonparametric Bayesian Methods in Robotic Vision

12 Lei Pi (UL), External Knowledge Absorption in Chinese SMEs

13 Bob R. Schadenberg (UT), Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning

14 Negin Samaeemofrad (UL), Business Incubators: The Impact of Their Support

15 Onat Ege Adali (TU/e), Transformation of Value Propositions Into Resource Re-Configurations Through the Business Services Paradigm

16 Esam A. H. Ghaleb (UM), Bimodal Emotion Recognition From Audio-Visual Cues

17 Dario Dotti (UM), Human Behavior Understanding From Motion and Bodily Cues Using Deep Neural Networks

18 Remi Wieten (UU), Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks

19 Roberto Verdecchia (VU), Architectural Technical Debt: Identification and Management

20 Masoud Mansoury (TU/e), Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems

21 Pedro Thiago Timbó Holanda (CWI), Progressive Indexes

22 Sihang Qiu (TUD), Conversational Crowdsourcing

23 Hugo Manuel Proença (LIACS), Robust Rules for Prediction and Description

24 Kaijie Zhu (TUE), On Efficient Temporal Subgraph Query Processing

25 Eoin Martino Grua (VUA), The Future of E-Health Is Mobile: Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications

26 Benno Kruit (CWI & VUA), Reading the Grid: Extending Knowledge Bases From Human-Readable Tables

27 Jelte van Waterschoot (UT), Personalized and Personal Conversations: Designing Agents Who Want to Connect With You

28 Christoph Selig (UL), Understanding the Heterogeneity of Corporate Entrepreneurship Programs

---

2022 01 Judith van Stegeren (UT), Flavor Text Generation for Role-Playing Video Games

02 Paulo da Costa (TU/e), Data-Driven Prognostics and Logistics Optimisation: A Deep Learning Journey

03 Ali el Hassouni (VUA), A Model a Day Keeps the Doctor Away: Reinforcement Learning for Personalized Healthcare

04 Ünal Aksu (UU), A Cross-Organizational Process Mining Framework

05 Shiwei Liu (TU/e), Sparse Neural Network Training With in-Time Over-Parameterization

06 Reza Refaei Afshar (TU/e), Machine Learning for Ad Publishers in Real Time Bidding

07 Sambit Praharaj (OU), Measuring the Unmeasurable? Towards Automatic Co-Located Collaboration Analytics

08 Maikel L. van Eck (TU/e), Process Mining for Smart Product Design

09 Oana Andreea Inel (VUA), Understanding Events: A Diversity-Driven Human-Machine Approach

10 Felipe Moraes Gomes (TUD), Examining the Effectiveness of Collaborative Search Engines

11 Mirjam de Haas (UT), Staying Engaged in Child-Robot Interaction, a Quantitative Approach to Studying Preschoolers' Engagement With Robots and Tasks During Second-Language Tutoring

12 Guanyi Chen (UU), Computational Generation of Chinese Noun Phrases

13 Xander Wilcke (VUA), Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented Knowledge

14 Michiel Overeem (UU), Evolution of Low-Code Platforms

15 Jelmer Jan Koorn (UU), Work in Process: Unearthing Meaning Using Process Mining

16 Pieter Gijsbers (TU/e), Systems for AutoML Research

17 Laura van der Lubbe (VUA), Empowering Vulnerable People With Serious Games and Gamification

18 Paris Mavromoustakos Blom (TiU), Player Affect Modelling and Video Game Personalisation

19 Bilge Yigit Ozkan (UU), Cybersecurity Maturity Assessment and Standardisation

20 Fakhra Jabeen (VUA), Dark Side of the Digital Media - Computational Analysis of Negative Human Behaviors on Social Media

21 Seethu Mariyam Christopher (UM), Intelligent Toys for Physical and Cognitive Assessments

22 Alexandra Sierra Rativa (TiU), Virtual Character Design and its potential to foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations

23 Ilir Kola (TUD), Enabling Social Situation Awareness in Support Agents

24 Samaneh Heidari (UU), Agents with Social Norms and Values - A framework for agent based social simulations with social norms and personal values

25 Anna L.D. Latour (LU), Optimal decision-making under constraints and uncertainty

26 Anne Dirkson (LU), Knowledge Discovery from Patient Forums: Gaining novel medical insights from patient experiences

27 Christos Athanasiadis (UM), Emotion-aware cross-modal domain adaptation in video sequences

28 Onuralp Ulusoy (UU), Privacy in Collaborative Systems

29 Jan Kolkmeier (UT), From Head Transform to Mind Transplant: Social Interactions in Mixed Reality

30 Dean De Leo (CWI), Analysis of Dynamic Graphs on Sparse Arrays

31 Konstantinos Traganos (TU/e), Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management

32 Cezara Pastrav (UU), Social simulation for socio-ecological systems

33 Brinn Hekkelman (CWI/TUD), Fair Mechanisms for Smart Grid Congestion Management

34 Nimat Ullah (VUA), Mind Your Behaviour: Computational Modelling of Emotion & Desire Regulation for Behaviour Change

35 Mike E.U. Ligthart (VUA), Shaping the Child-Robot Relationship: Interaction Design Patterns for a Sustainable Interaction

2023 01 Bojan Simoski (VUA), Untangling the Puzzle of Digital Health Interventions

02 Mariana Rachel Dias da Silva (TiU), Grounded or in flight? What our bodies can tell us about the whereabouts of our thoughts

03 Shabnam Najafian (TUD), User Modeling for Privacy-preserving Explanations in Group Recommendations

04 Gineke Wiggers (UL), The Relevance of Impact: bibliometric-enhanced legal information retrieval

05 Anton Bouter (CWI), Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization, Including Real-World Medical Applications