Vrije Universiteit Amsterdam Universiteit van Amsterdam

Master Thesis

Explaining the Explainer

Author: Willem van der Spek (2607407)

1st supervisor:	Adam Belloum	
daily supervisor:	Elena Ranguelova	(Netherlands eScience Center)
2nd reader:	Rob van Nieuwpoort	

A thesis submitted in fulfillment of the requirements for
the joint UvA-VU Master of Science degree in Computer Science

September 28, 2023

"As far as we can discern, the sole purpose of human existence is to kindle a light in the darkness of mere being."

from Memories, Dreams, Reflections*, by Carl Gustav Jung*

Abstract

In recent years, *Explainable Artificial Intelligence* (XAI) has emerged as a critical field in bridging the gap between complex black-box models in *Machine Learning* (ML) and human comprehension. A pressing challenge that is hindering the adoption of XAI is the absence of quantitative evaluation methods for explanations. In other words, how do we measure the quality of an explanation? Furthermore, there's a lack of a systematic approach for selecting the right XAI algorithms and fine-tuning their hyperparameters.

In response to these challenges, our work incorporates several quantitative evaluation metrics for XAI based on recent advancements in the field. These metrics are designed to assess two crucial aspects of explanations: *correctness* and *continuity.* We argue that reliable explanations must accurately represent the underlying model and provide consistent insights for similar instances. Using these metrics, we conduct two extensive case studies, encompassing both image and textual data. In these studies, we evaluate explanations generated by a variety of XAI algorithms based on the proposed metrics.

Our main findings offer essential insights into the impact of hyperparameters on explanation quality. Furthermore, we compare the quality of explanations across different XAI algorithms, considering both image and text data.

Contents

List of Figures

List of Tables

1

Introduction

Machine Learning (ML) models have become ubiquitous over the past decade. However, the more complex ML models, such as Deep Learning (DL) models display a lack of interpretability. This problem has been dubbed as the *black-box* problem in the field of ML, where models have become too complex to be interpreted by humans. In spite of this, models are still becoming increasingly complex due to their general tendency to produce better predictive accuracy.

In response to this, a new field has emerged: *eXplainable Artificial Intelligence* (XAI), which has experienced a meteoric rise in the past few years (27). The ultimate goal of XAI is to mitigate the black-box problem by explaining decision processes in ML. In doing so, researchers hope to achieve a variety of desiderata for AI, namely: improving end-user trust in ML algorithms, producing more explainable models with only marginal effect on the ML model's performance, improving the ML model development process by providing insights in model-data interactions, enhancing model fairness and supporting a variety of data modalities and models, just to name a few.

Moreover, a large variety of fields have started to embrace XAI with promising prospects. Meteorologists have been using machine learning to predict droughts with Dikshit and Pradhan (13) incorporating XAI as a means to enhance trust in this community. With ML moving to the medical field, XAI could help in providing more transparency in making clinical decision (22). The field of criminal justice, moreover, could incorporate XAI to unveil racial biases in recidivism prediction (16). Emerging fields such as autonomous driving could also greatly benefit from increased transparency as such fields still harbor a lot of public skepticism and legislative barriers (34).

Thus, it is of crucial importance that the explanations generated by XAI provide a level of reliability such that relevant stakeholders are provided with credible information about

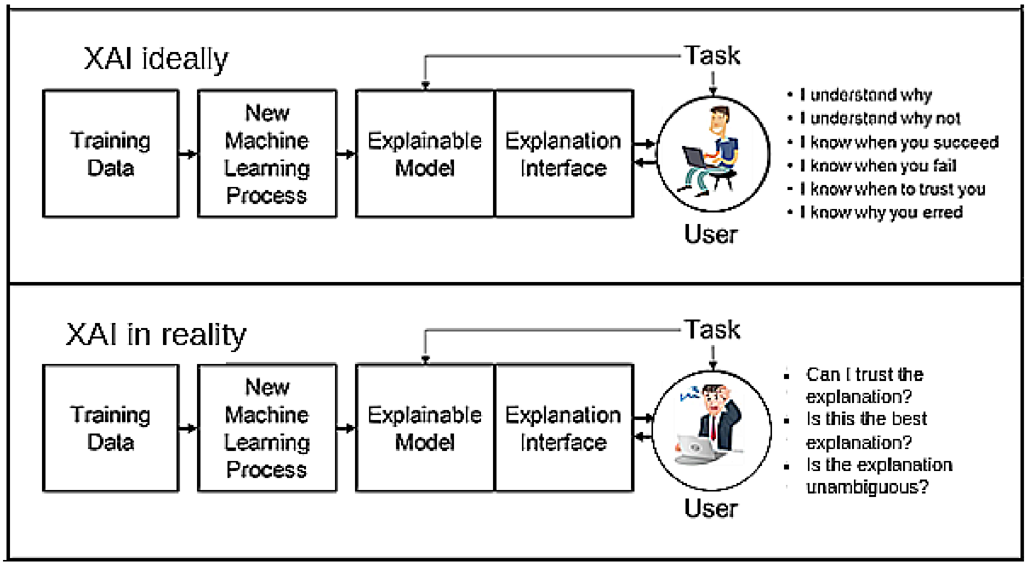


Figure 1.1: Users still remain sceptical of explanations generated by XAI algorithms, this is a key issue for wider adoption of XAI. Image inspired by the work of Gunning and Aha (21).

the models being explained. Still, some issues of transparency remain between XAI itself and its users. XAI algorithms are known to exhibit weaknesses, including sensitivity to adversarial attacks, where relatively small input perturbations could completely alter explanations in both arbitrary (19) and targeted ways (14). Some other works have criticised a multitude of XAI algorithms by means of *sanity checks for salience maps*, where explanations are tested against completely untrained models to prove if they are actually describing the model. Shockingly, the behaviour of several XAI algorithms was found to be independent of the model that was to be explained (1). In general, these concerns undermine the transparency of XAI itself which we have illustrated in figure 1.1.

More recent research trends have started incorporating evaluation practices for XAI to ensure the explanations of the model are of satisfactory quality. It is worth noting that these evaluations can be done either subjectively by humans or automatically with evaluation metrics (37). Additionally, these metrics should describe several facets of explanations which in turn describe the quality of the explanation. Even though the establishment of a consensus on these metrics remains an ongoing issue, the most recent advancements suggest that the field of metrics for XAI is now maturing with an increasing volume of works incorporating some form of quantitative evaluation in their work (37). Concurrently, some software packages introducing a standardised approach to evaluation are being established (2, 24).

Aside from providing objectively guarantees to explanations, metrics could further advance the field in several new other ways. Another issue in the field is a lack of a principled approach to hyperparameter selection in XAI. The effects of hyperparameters on explanations remain largely in obscurity. Some theoretical guides to hyperparameter tuning were proposed by Vermeire et al. (51) and an approach to optimisation by Cugny et al. (11). Yet, to the best of our knowledge, a thorough study on these hyperparametrisations appears to be lacking.

1.1 Research Questions & Approach

This work will focus on and investigate the effect of several hyperparameter configurations on explanation quality using a variety of XAI algorithms in an experimental setting through a set of case studies and quantitative evaluation metrics for XAI. We present our central research questions as follows:

How do XAI algorithms and their hyperparametrisations affect explanation quality?

We further decompose this in sub-questions (sQ):

sQ1 How can explanation quality be defined qualitatively?

sQ2 How can explanation quality be quantified?

sQ3 What is the relationship between individual hyperparameters and explanation quality?

sQ4 To what extent do explanations benefit from optimisation?

sQ5 How do explanation methods compare in terms of explanation quality given optimal hyperparameters?

1.2 Thesis outline

The remainder of this thesis is organized as follows:

Chapter 2 represents the necessary terminology and background paramount to understanding our work.

Chapter 3 represents a set of works that are related in goal and approach and will argue for the novelty and contribution of this work.

In chapter 4 we discuss and motivate the technical design choices made in order to answer our research questions.

Chapter 5 proceeds to show the results achieved from our experimental design along with some direct observations.

In chapter 6 we interpret the results in relevant context and take a critical perspective to our results with regard to the design choices.

Finally, chapter 7 contains a summarisation of our findings and provides suggestions for future work.

1.3 Contributions

Our technical contributions can be found online on Github [1], we summarise these contributions are as follows:

1) Developed the `Incremental Deletion` metric for image data.

2) Developed the `Single Deletion` metric for textual data.

3) Adjusted code from Quantus (24) to accomodate for `Sensitivity` metric for textual data.

4) Provided tutorials for all metrics, including the ones from Quantus.

5) Developed a configurable grid for the hyperparameters in DIANNA (43).

6) Developed a script to compute XAI metrics on the explanations using grid search.

7) Developed code for the dataset sampling

8) Developed code to analyse and visualise the metrics over the hyperparameter grid.

[1] Willem van der Spek, dianna-exploration, `https://github.com/dianna-ai/dianna-exploration/tree/main/relevance_maps_properties`

2

Background

This chapter provides the necessary theoretical knowledge in order to get acquainted with the field. It includes a brief introduction to the field through a popular taxonomy, in-depth description of the algorithms used and describes the problem of evaluating explanations in XAI.

2.1 Taxonomic Overview of XAI

Multiple taxonomies have been proposed to conceptualise the large amount of XAI algorithms that are being proposed. Speith (49) have conducted a review of taxonomies in XAI, and suggest the taxonomy shown in figure 2.1. The purpose of this taxonomy is to (1) provide a holistic snapshot of the algorithmic landscape of XAI and (2) discuss each dimension of abstraction in more detail in order to provide an introduction to the field. Some of the dimensions will be excluded, such as the *Output Format*, which we assume to be self-evident. The *functioning* dimension will be left out as well as the functioning of XAI algorithms relevant to this paper will be extensively covered in section 2.2. All of these algorithms rely on *input perturbations*, which aim to find feature relevances through randomly perturbing input instances.

2.1.1 Scope

The *scope* refers to the locality of the data that is to be explained.

2.1.1.1 Global

Global explanations refer to explanations based on the entire dataset relevant to the black-box model. Global explanations provide general insights on the interactions between the

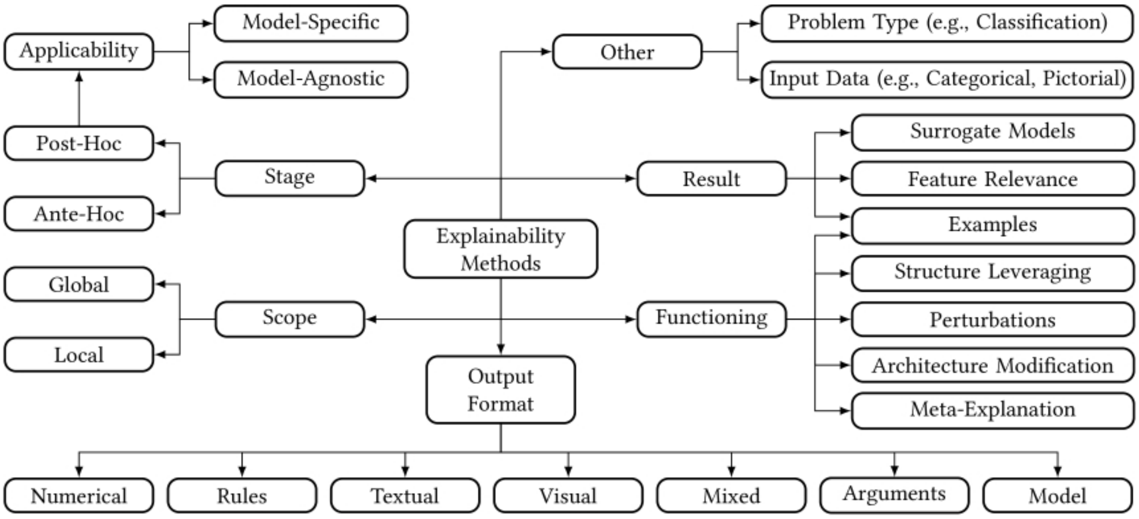


Figure 2.1: Taxonomy of explainability methods proposed by Speith (49).

data and the model and are particularly useful in uncovering patterns that hold true in the entire dataset. Examples are explaining the roles of gender, race, age or disabilities in recidivism (4).

2.1.1.2 Local

On the contrary, *local explanations* concern a particular *data instance*. Local explanations are valuable when you want to understand the reasoning behind individual model predictions, which can be relevant to capture nuanced insights and address specific cases. An example would be to explain why a gender detection model marks an image of a face as male or female.

2.1.2 Stage

This taxonomy dimension refers to the processing stage; the time at which explainability is introduced into the black-box model.

2.1.2.1 Ante-hoc

Ante-hoc explanations are those that are incorporated into the model's design and training process from the very beginning. In other words, the model is engineered with interpretability in mind right from its inception. The goal of ante-hoc explanations is to ensure that the model remains transparent and understandable throughout its life-cycle. More traditional machine learning methods, such as linear regression and decision trees fall under ante-hoc methods due to their inherent interpretability (e.g. the information gain in decision trees and the coefficients for linear regression).

2.1.2.2 Post-hoc

On the contrary, *post-hoc explanations* are generated after the training process of a black-box model. In this approach, the inner workings of the model will remain opaque, with the goal being to devise a method in order to provide an explanation in a retrospective manner. Post-hoc methods are more popular than their ante-hoc counterparts likely due to the preservation of the black-box model and its advantages in terms of predictive accuracy (52).

2.1.3 Applicability

Some XAI methods are specifically designed for a class of ML models, while others are generic. As such, *applicability* refers to the range of models the XAI algorithm covers, and more specifically, its need to access the model's internals.

2.1.3.1 Model-specific

Model-specific algorithms aim to leverage the inner workings of a black-box model into an explanation. These techniques can take into account several properties of the black-box model, such as their architecture, features and decision processes. As such, these algorithms are typically limited to a specific model. Marked examples of model-specific algorithms include Integrated gradients (50) and GRAD-CAM (46). Each of these methods require some leveraging of the model architecture in order to arrive at an explanation. Model-specific methods come with the advantage of being able to be customised for a specific model and the ability to leverage extra information contained within the model architecture (2).

2.1.3.2 Model-agnostic

On the contrary, *model-agnostic* methods generate explanations without requiring any knowledge about the internals of the model. The black-box is merely regarded as a function (say $f(x)$) which output is used to generate explanations. As a result, model-agnostic methods typically have the advantage of being applicable to a large variety of models than their model-specific counterparts.

2.1.4 Result

The dimension of *result* concerns itself with the presentation of the explanation. The distinction is made between *surrogate models*, *examples* and *feature relevance.* The algorithms covered in this work will mainly be concerned with feature relevance, the idea of which is to break down model input x into its individual features x_i and give relative scores to each of these features, i.e. feature relevances. The result is then a clear indication of which individual aspects of the data are driving the model's decision. The term *feature relevance* is used interchangeably with *salience/attribution map*, *feature importance/attributions* in this work.

Surrogate models can be used as an alternative model that captures the key decision-making aspects of the original model in a more understandable way. Surrogate models can also be used to obtain feature relevances, as is the case for *Locally Interpretable agnostic Model Explanations* (LIME), described in section 2.2.2

Examples involve carefully selecting representative instances that illustrate how the model works. The result is a set of real-world examples that showcase the decision-making process of the model.

2.2 XAI Algorithms

This section will cover in-depth descriptions of the XAI algorithms used in this work. Considering the taxonomy in figure 2.1, the dimensions these XAI algorithms encompass are *Local* for *Scope*, *Post-Hoc* for *Stage*, *Model-agnostic* for *Applicability*, *Surrogate Models* and *Feature Relevance* for *Result* and *Perturbations* for *Functioning.* Furthermore, the algorithms covered in this section will be subject to a set of *hyperparameters.* We have provided an overview of the hyperparameters for each of these algorithms in table 7.1.

2.2.1 RISE

Randomized Input Sampling for Explanations (RISE) was introduced as an XAI algorithm in 2018 (42). It was originally proposed as a means to obtain feature relevances for image data, though RISE can be extended to operate on different data modalities as well. Being model-agnostic, RISE doesn't require access to the model's internals and has a straightforward approach for obtaining feature relevances. Furthermore, RISE comes with the advantage of working at the lowest level of feature resolution (e.g. pixels for image and single words for textual data), and attributes feature relevances to individual features, whereas other popular XAI algorithms require feature segmentation.

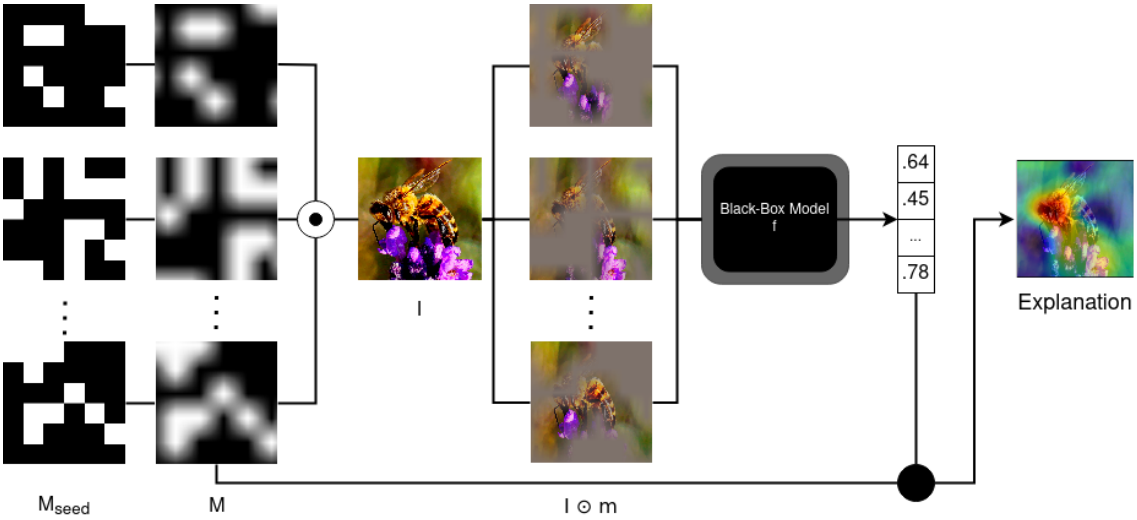


Figure 2.2: A visual representation of the RISE algorithm inspired by the original paper (42). First n seed masks are generated with distribution $\mathscr{D}$. These seed masks are then upsampled, typically through bi-linear upsampling to match the image dimensions. Afterwards the image is masked with the upsampled mask producing the perturbed instances seen in the fourth column. These perturbed instances are used to feed through the model, obtaining probabilistic vectors for each of the classes. Using a weighted sum, the masks and their appurtenant probability vectors are aggregated to obtain feature relevances for each pixel.

Figure 2.2 provides a visual representation of the RISE algorithm. RISE leverages a set of randomly generated masks that perturb features in a given instance and consecutively measures the effect of these perturbations on the model scores. These perturbations involve making relatively small changes to input features. The core intuition behind RISE is that some features are more strongly correlated to the model score than others and the observation of this effect can be used to compute feature attributions. When dealing with image data, the mask space grows by a factor of $2^{H \times W}$ with H and W being the height and width of the image. Finding a representative set of masks in such a large space is therefore computationally unfeasible in almost all cases. In order to address these issues, RISE uses *bilinear upsampling* on a set of smaller masks to obtain a more representative set of masks. This technique uses repeated linear interpolation on an input image which is in turn is used to extrapolate beyond the input mask (M_{seed}), yielding an upscaled image. This process produces particularly smooth edges, which results in a smoother explanation. It must be noted that for textual data, this bilinear upsampling is not possible, hence one needs to directly mask text without any data transformation process. Finally, RISE normalises it values by the number of masks n and the expected value of the masks $\mathbb{E}[M]$.

Formally, RISE can be expressed as follows. Given a black-box model f and an input instance I, compute feature relevances S. Let M be the set of all possible random masks that applies to I, these masks are generated through the bilinear upsampling of masks M_{seed} with dimensions $r \times r$ with r being the Resolution parameter (see table 7.1). Given that the number of possible subsets in M_seed becomes virtually impossible to compute, RISE uses Monte-Carlo sampling to estimate its feature relevances with a total of n masks:

$$S_{I,f} \overset{\text{MC}}{\approx} \frac{1}{\mathbb{E}[M] \cdot n} \sum_{i=1}^{n} f(I \odot M_i) \cdot M_i \tag{2.1}$$

2.2.2 LIME

Locally Interpretable agnostic Model Explanations (LIME) is another model-agnostic approach to model interpretation, introduced by Ribeiro et al. (44) in 2016. In similar fashion to RISE, LIME doesn't have to consider the model's internals and can be applied to a multitude of data modalities. Moreover, input perturbations are used in order to observe the effect between relevant features and model score.

Contrary to RISE, LIME uses *surrogate modeling* to produce explanations, which is a simpler, interpretable model (surrogate) that approximates the behavior of the complex model within a certain locality. Typically, LIME employs linear regression as the surrogate model due to its simplicity and ease of interpretation. Linear models can be directly interpreted though their coefficients which indicate the importance of distinct features. It's important to note that the LIME perturbations are generated in the neighbourhood of the instance and are designed to be *locally faithful*, i.e. these explanations adhere strongly to the data in the neighbourhood but are not guaranteed to work outside of this local neighbourhood. Additionally, for image data, LIME requires *image segmentation* (i.e. finding relevant regions in the image) in order to function properly.

Figure 2.3 serves as a geometric interpretation of the algorithm using a synthetic dataset with samples drawn from two interleaving circles.

Formally, the process of finding an appropriate surrogate model $\Theta(x)$ is defined in equations 2.2 and 2.3, where G represents the set of all possible models subject to loss function $\mathscr{L}$. Ω is a measure of complexity for a model, less complex models, e.g. linear models with more zero coefficients are preferred. Thus, finding the best surrogate model includes minimizing $\mathscr{L}$ and Ω.

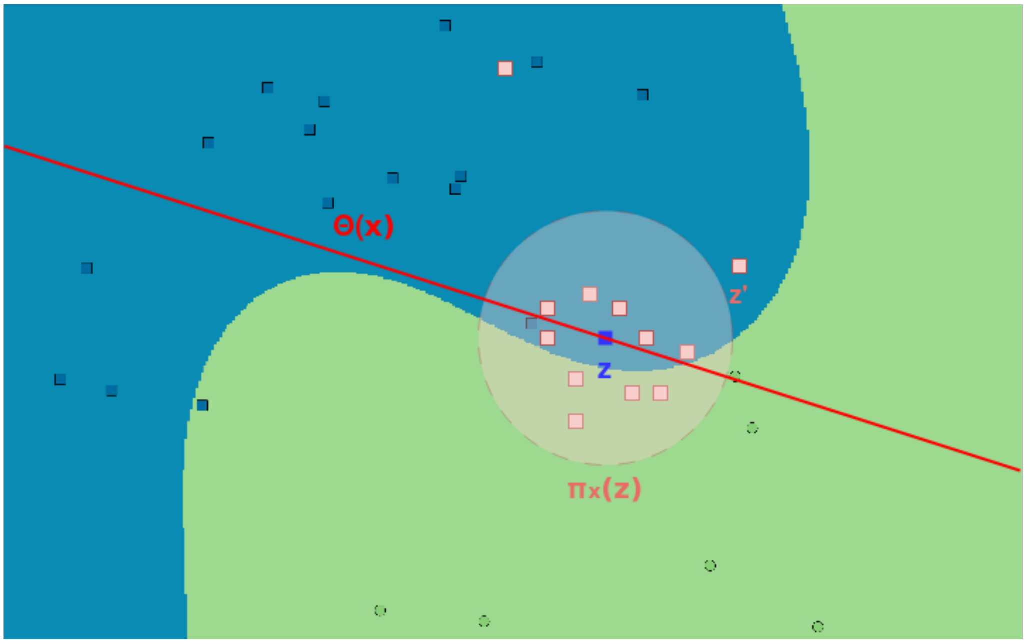


Figure 2.3: Geometric interpretation of LIME using a toy dataset as example. The green and blue regions indicate the *decision boundaries* for the black-box model while the the blue squares and green dots represent separate classes. An instance z is explained by generating perturbations z' in its neighborhood by querying the model. These perturbations are used to fit a linear model $\Theta(x)$ with a penalization by means of distance kernel π_x.

$$\Theta(x) = \arg\min_{g\in G} \mathscr{L}(f, g, \pi_x) + \Omega(g) \tag{2.2}$$

Loss function $\mathscr{L}$ is typically defined as a locally weighted version of the *ordinary least squares* algorithm and is defined in 2.3 below. The weight for the instances is given by $\pi_x(z)$ where an exponential kernel is typically used: $\pi_x(z) = \exp(-D(x,z)^2/\sigma^2)$ with some distance function D and kernel width x. The behaviour of the black-box model is introduced by the $f(z) - g(z')$ part which aims to minimize the discrepancy between the opaque model and black-box model.

$$\mathscr{L}(f, g, \pi_x) = \sum_{z,z'\in\mathscr{Z}} \pi_x(z)(f(z) - g(z'))^2 \tag{2.3}$$

It is worth noting that LIME is subject to a large amount of hyperparameters, the ones listed in table 7.1 only scratch the surface. For example, producing satisfactory explanations requires a representative sample of perturbations which in turn is dependent on the distance kernel and its distance metric, kernel width and the number of perturbations. Additionally, using LIME on image data requires segmentation of the image in order to discretise its features which further adds to the complexity of optimising LIME explanations.

2.2.3 KernelSHAP

SHapley Additive exPlanations (SHAP) is based on the idea of *Shapley regression values*, which were originally introduced by Lipovetsky and Conklin (32) in 2001. Lundberg and Lee (33) have built upon this work to compute these Shapley values in the context of predictive modelling. SHAP relies on a game-theoretic approach to model interpretability, selecting combinations of features and comparing their relative effects with the exclusion of targeted features. Exact Shapley values take exponential time to compute, and typically, estimations of these values are used in practice. KernelSHAP is a means of estimation of Shapley values through a combination of LIME and SHAP. Similar to LIME, KernelSHAP requires the usage of segmentation for image data.

Equation 2.4 shows the key formula to obtain the Shapley values ϕ_i which involves considering all possible subsets of features for instance x denoted by z'. For each of these subsets, feature i is discarded and the difference between the model effect on these subset is computed by $f_x(z') - f_x(z' \setminus i)$. This difference is then weighted and summed across the sampled subsets to exactly obtain the Shapley value for each feature.

In the equation $\phi_i(f, x)$ represents the Shapley value of feature i for a specific instance x and model f. The contribution of feature i is determined by the difference between the model's prediction using a subset of features z' and the prediction when excluding feature i from that subset $z' \setminus i$. Aggregating these results together through a weighted sum consequently will yield the Shapley value for i.

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)] \tag{2.4}$$

We have already briefly discussed that the number of all possible subsets of the set of features is $O(2^M)$ with M being the number of features in the dataset. Hence, the authors have proposed KernelSHAP as a method to estimate the Shapley values instead. KernelSHAP aims to consider the most relevant combinations of features by estimating them through a linear kernel (LIME).

$$\begin{aligned}
\Omega(g) &= 0 \\
\pi_{x'}(z') &= \frac{(M-1)}{(M \text{ choose } |z'|)|z'|(M-|z'|)} \\
\mathscr{L}(f, g, \pi_{x'}) &= \sum_{z' \in \mathbb{Z}} [f(h_x^{-1}(z')) - g(z')]^2 \pi_{x'}(z')
\end{aligned}$$

This set of equations is similar to LIME. The distance kernel $\pi_{x'}(z')$ is used to penalise subsets that are further from 0 or the total number of features M by the term M choose $|z'|$ which denominates the binomial coefficient of M over $|z'|$. This weighting emphasises inclusion of a small number of features (because it highlights the "independent behavior of features) or almost all of them (because it highlights the impact of features in interaction with all of the others).

The final term computes a weighted square loss between black-box model f and surrogate model g. KernelSHAP introduces one more nuance which is $h_x^{-1}(z')$, which takes a transformed input z' and maps it back to the original input space x. This is important because the process of obtaining Shapley values with KernelSHAP involves generating samples from the conditional distribution.

2.3 Explanation Evaluation

2.3.1 Explanation Properties

Hitherto, we have discussed XAI simply as means to generate model explanations. However, in order to enable trust in explanations by actual users, explanations must exhibit some degree of credibility. As such, a volume of research works have defined several *explanation properties*, which in turn reflect explanation desiderata (i.e. what do we want out of explanations?). We have compiled a list of these explanation properties in table 2.1, following the work of Nauta et al. (37). It is worth noting that different user groups have different desiderata, which requires a modularised scheme of explanation properties, such as the one provided by Sokol and Flach (48). In line with this, several studies have outlined distinct XAI user categories (8) : *lay users*, *domain experts* and *AI experts. Lay users* are users who are considered to have little to no knowledge about neither the data domain, nor the model development process. Contrary, the *Domain Experts* are considered to have strong knowledge about the data domain the model is operating on, but not necessarily about the model development. Finally, the *AI Experts* are well aware of the model development process. These users are typically interested in XAI as a means to gain insights into the model-data interactions and improving their model. This consideration of *target audience* is crucial when compiling a set of explanation properties; explanation quality is dependent on its desiderata, which in turn depend on target audience. Our target audience are the Domain Experts.

Table 2.1: Explanation properties from the work of Nauta et al. (37). Each explanation property reflects a part of explanation quality as a whole.

Property	**Description**
Correctness	This property emphasizes the accuracy of explanations. An explanation is considered correct when its behaviour is faithful to the black-box model it explains.
Output-Completeness	The property of completeness pertains to the extent to which explanations encompass all relevant aspects of the decision. Complete explanations guarantee the "whole truth".
Consistency	Consistency describes the determinism of explanation methods. It ensures that the exact same instances should have explanations that are either exactly the same or marginally dissimilar.
Continuity	Continuity discerns the robustness of the explanation function against small input perturbations and in turn the smoothness and generalisability of the explanation function. Explanations that exhibit high continuity are considered to be more resilient against *adversarial attacks*.
Contrastivity	Describes how discriminate the explanation is with regard to other targets.
Covariate Complexity	Describes how complex the interactions between features in the explanation are.

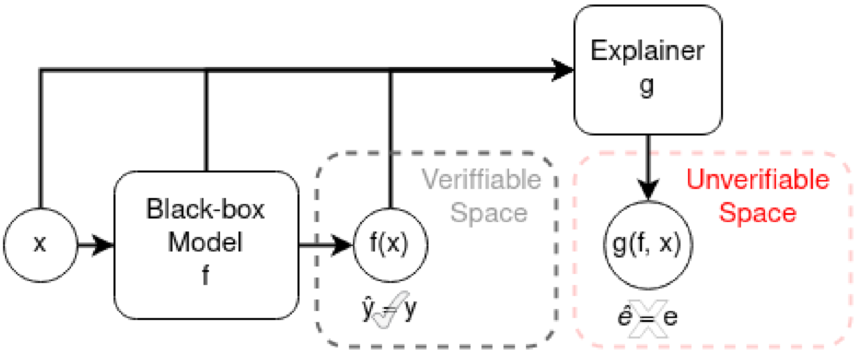


Figure 2.4: The core problem in XAI evaluation. Explanations are projected onto an unverifiable space without the presence of an unambiguous ground truth, complicating the process of evaluation. Inspired by Hedström et al. (25).

2.3.2 Evaluation Metrics

Paradoxically, whereas standard evaluation metrics for evaluating the predictive performance of machine learning are ubiquitous, there is no consensus for evaluation practices in XAI. The crux of evaluating explanations lies in the absence of ground truth; if we were to know the ground truth of an explanation in advance, there would be no need for XAI in the first place. As a result, common evaluation practices include showing examples that look reasonable to the user (15). Furthermore, evaluating explanations as a function of its plausibility and persuasiveness to humans could be misleading; a model decision might not align with human intuition and domain knowledge. Unreasonable explanations could both be the result of erroneous reasoning obtained from the black-box model or the explanation method (20). Using visual inspection to judge explanations could lead to the spurious coupling of the predictive accuracy of the black-box model and the explanation quality leading to biased evaluations (8, 30). We have illustrated this problem more generally in figure 2.4, which shows that supervised machine learning operates in a verifiable space (all elements have an unambiguous ground truth), whereas explainability methods use the supervised machine learning pipeline to arrive at the explanation, which is part of the unverifiable space (there is no ground truth).

As a result of these concerns, a variety of *quantitative evaluation metrics* have arisen in the field. Contrary to metrics in ML, these are far more complex and require integration of the input data, black-box model and explanation. Each of these metrics propose to quantitatively evaluate explanations by factoring in information from both the black-box model as well as its explanation. Table 4.1 provides an overview of the metrics used in this work and sections 4.6.1, 4.6.2 and 4.6.3 will cover these metrics in depth.

3

Related Work

This chapter focuses on works related to this work in terms of approach, goal and design. It presents the necessary information to conceptualise its position in the necessary fields of research that were considered and argues how this work contributes and what the novelties of this work are.

3.1 General position in the field

This work is at the intersection of several research topics, namely XAI Algorithms, quantitative evaluation for XAI and XAI hyperparameter optimisation. A variety of XAI algorithms have been proposed by the research community and to such an extent that it requires an extensive taxonomy to encapsulate them. Additionally, a variety metrics have been proposed by the research community, albeit without a general consensus on them (37).

The disregard for quantitative evaluation in XAI cannot be understated, and works that utilise quantitative evaluation in real-world settings help in pushing the field forward by showing its applicability. This disregard for evaluation practices perpetuates to other fields in XAI as well. Hyperparameter optimisation is, as a result, even more understudied as it requires evaluation in the first place. Only several works have attempted to optimise hyperparameters in XAI, hence our list of related works is relatively short. We briefly elaborate on the few works that were similar in approach and research goal to this work. We argue that our work has provided a valuable contribution to the field by yielding relevant new insights for XAI hyperparametrisations, an approach to hyperparameter optimisation and the usage of quantitative evaluation metrics in that regard. Moreover, this work marks one of the studies that encompasses multiple modalities and one of the first approaches to evaluation metrics for textual data in XAI.

3.2 Descriptions of related works

The most closely related work to ours would be that of Cugny et al. (11). The authors provided automated approach to find optimal hyperparameters was used with variety of metrics. A ranking system based on user preferences to select optimal hyperparameters using *Bayesian optimisation* was proposed. The work presented a system that could be used to optimise explanations with promising results. However, what the work did not consider was a cross-algorithmic comparison and insights on how the hyperparameters affected the explanations. Instead of providing a system for finding optimal hyperparameters, this approach aims to capture several relationships between the parameters and the metrics. Through this approach, we have provided relevant insights on interactions between hyperparameters and metrics and consequently allow for a more principled approach on hyperparameter selection. Besides, we included a comparison between algorithms in terms of explanation quality. Finally, this work is vastly different in setting, where different XAI algorithms, black-box models and datasets were used.

Visani et al. (54) have conducted work on gathering insights on the hyperparameters and their effects on explanation quality for LIME specifically. Interestingly, this work was already able to capture some relationships for LIME. The authors concluded that the ridge penalty for the surrogate model tends to be harmful for the stability of the explanation and proposed simple linear regression as the surrogate model. Furthermore, kernel width and *correctness* or *local faithfulness* were found to be inversely proportional, whilst *consistency* was found to be proportional to the explanation. Nevertheless, this work is limited to a single algorithm - LIME - with metrics specifically designed for this algorithm. Our work has proposed a variety of XAI algorithms and different data modalities whilst using a variety of metrics.

Bansal et al. (7) have investigated the sensitivity of hyperparameters with regard to the explanation by varying a set of hyperparameters per XAI algorithm for image data. Their experiments were conducted on image data and included enhancing models in order to make them more robust to adversarial attacks. Adversarial attacks try to manipulate explanations (or ML predictions) by making visually imperceivable alterations to the input data. The findings suggested that XAI algorithms were highly sensitive to the choice of hyperparameters. Additionally, it was concluded that robust models generate explanations that are more robust as well. In similar trend, this work has produced findings on the sensitivity of hyperparameters but has expanded by investigating additional algorithms and hyperparameters. Additionally, this work has studied the relationships between individual hyperparameters and their effect on explanation quality.

Pahde et al. (40) have similarly evaluated explanations and tried to optimise its hyperparameters through quantitative evaluation metrics. Specifically, grid search was used to try out several predefined configurations aimed to find optimal configurations. The authors verified a total of 2592 configurations for the γ parameter in *Layerwise Relevance Propagation* (LRP) (31), another *post-hoc* XAI algorithm. The authors evaluate this effect Our work does not introduce a new approach to the ML pipeline or explanations. Rather, it is more straightforward and simply looks at the XAI algorithms and their effect on explanation quality.

Finally, Arras et al. (5) have devised a benchmark for XAI algorithms while simultaneously investigating the effect of hyperparametrisations on explanation quality. Using grid search, the authors came to a configuration of optimal hyperparameters for each of their methods and selected these in order to compare XAI algorithms according to their proposed benchmarks. These benchmarks are based on ground truth of explanations that the authors generated using an automated approach. On the contrary, this work aims to avoid the usage of ground-truth masks and focuses on metrics that cover *correctness* and *continuity* instead.

4

Experimental Design

This chapter provides the core of the research. We present an overview of the experiments and choices made with respect to its components. These design choices are what we used to obtain our results in the following chapter.

4.1 Overview

In order to assess the quality of explanations under varying hyperparametrisations we have devised experiments with combinations of quantitative evaluation metrics, XAI algorithms, data modalities, models and hyperparameter configurations. For a high-level overview of our experimental design, we refer to figure 4.1. The general idea and purpose of this experimental design was to capture the effect of hyperparametrisations of an explainer on explanation quality across a variety of different experimental settings. In order to measure explanation quality, a set of quantitative evaluation metrics for XAI were selected, for which we have provided an overview in table 4.1. From the explanation qualities as described in table 2.1, we have decided to measure *Correctness* and *Continuity.* For our user group of *Domain Experts*, who require robust explanations that represent the underlying black-box model well. As such, these properties were deemed as more important than the other ones for evaluation. Moreover, we measure the runtime of our explanations. The remainder of this chapter will provide more in-depth explanations and motivations for the multitude of choices made for our experimental design, including the datasets used, data sampling strategy, black-box models, implementation of XAI algorithms, hyperparameter configurations for the explainers and the quantitative evaluation metrics used to evaluate explanations.

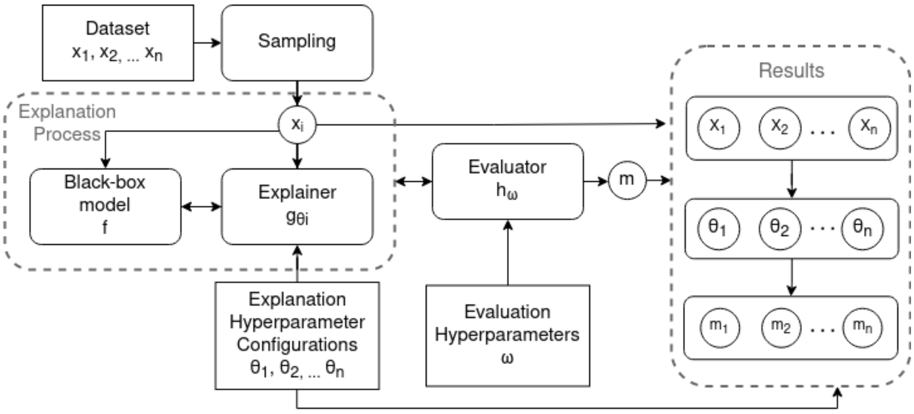


Figure 4.1: A high-level overview of our Experimental design. The final results are quantitative evaluation metrics m grouped by hyperparameter configuration θ_i and data instance x.

Table 4.1: Quantitative evaluation metrics chosen for our experiments along with their desired *property* formula and a short description.

Property	Metric	Formula & Description
Correctness	Incremental Deletion (42, 45)	$\text{ID}(f, g, x) = \int_0^n f(x_{rand}^{(k)})dk - \int_0^n f(x_{MoRF}^{(k)})dk$ Incrementally remove features from an instance based on their importance given by the explanation. Compare with a random baseline.
Correctness	Single Deletion (28)	$\text{SD}(f, g, x) = \underset{x_i \in x}{\text{corr}}(g(f, x)_i, f(x) - f(x \setminus x_i))$ Delete single instances from the original data x_i. Compute correlations between the model score for the perturbed data and the feature attributions .
Continuity	Sensitivity (55)	$\text{SENS}(f, g, x, r) \overset{\text{MC}}{\approx} \frac{\|\|g(f,x+\epsilon,r)-g(f,x,r)\|\|_{\text{F}}}{\|\|g(f,x,r)\|\|_{\text{F}}}$ Through Monte Carlo sampling, perturb the instance n times and compute the differences between the original explanations and the explanations of the perturbed instances.

4.2 Definitions

We provide some definitions for the symbols used in 4.1. Across formulas: f represents a black-box model, g an explanation function, x a data instance and x_i a feature within that instance. Let $g(f, x)_i$ be a relevance in a feature map generated by g in the definition of Single Deletion. In the Sensitivity definition, let ϵ be some kind of noise that is bound to neighbourhood r. Finally we provide a recursive definition for x_{MoRF} (Most Relevant First) from the work of Samek et al. (45):

$$x^{(0)}_{MoRF} = x$$
$$\forall 1 \leq k \leq L : x^{(k)}_{MoRF} = d(x^{(k-1)}_{MoRF}, x_k)$$

where function d removes features from instance $x^{(k-1)}_{MoRF}$ at a specified feature x_k (e.g. a pixel, word or image segment). Similarly, x^{k}_{rand} works by deleting the pixels in random order instead of most relevant with d now selecting arbitrary values.

4.3 Datasets

4.3.1 Image

The *binary MNIST* dataset is an image dataset consisting of two classes, extracted from the original MNIST dataset (12). These two classes entail images that either represent the digit ’1’ or the digit ’0’. All of the instances are represented in grayscale and are 28 × 28 pixels in size. The training dataset consisted of 12665 instances of which 5923 belong to class ’0’ and the remaining 6742 represent class ’1’. The test set contained 2115 samples, 980 of which belong to class ’0’ and 1135 - to class ’1’.

4.3.2 Textual

The *Stanford Movie Review* dataset (47) is a textual dataset that compromises two classes, similar to the binary MNIST. These two classes entail negative and positive sentiments expressed in movie reviews. All instances are pieces of English text of arbitrary length with a total of 8544 training instances with approximately even class balance. The original labels were scores of 1 to 10, representing sentiment with 1 being the lowest possible sentiment and 10 being the most positive. These scores were reduced to two labels: ’negative’ with scores less then 5 and ’positive’ with scores greater than 6. No more than 30 reviews were included per movie.

4.3.3 Sampling

A sample of the original test set was considered in order to meet computational demands. For a sampling strategy, we opted to utilise the model performance. Quantitative evaluation metrics are probably sensitive to the model score and when applying XAI to instances in a real-world scenario, one would likely encounter a variety of different model performances (e.g. examples where the model generates confusing results). To attune to this, we chose to sample according to the model score, where we select instances based on their model score, leading to a uniform distribution of model scores. The sampling process was performed in greedy fashion, searching for an instance with the closest score in the linear space between the minimum and maximum possible model scores as we have described in equation 4.1. For the textual data, only sentences with more than 5 tokens (instances of a sequence of characters in the text grouped together as a useful semantic unit for processing) were considered. Furthermore, sentences containing '(', ')' or '-' were excluded as these symbols proved to be cumbersome for interactions between the tokeniser and XAI algorithms. Considering our sampling equation, we choose a space size n of 100 and our model scores f_{max} and f_{min} were 1.0 and 0.5 respectively yielding linear space L from which instances with neighbouring scores were selected. This process yielded a total of 94 images and 99 sentences as data points.

$$L = [(f_{min} + i * \frac{f_{max} - f_{min}}{n})]_{i=0}^{n} \tag{4.1}$$

4.4 Models

In this section we briefly describe the black-box models, considering their architecture and performance statistics in table 4.2

4.4.1 Images

The model for the binary MNIST classification task was a *Convolutional Neural Network* (CNN) developed by Meijer and Liu (35). The model consisted of two similar sequences of hidden layers: a convolution layer, a max pooling layer, a dropout layer and fully connected layer. These hidden layers used ReLU as their activation function whilst a logarithmic softmax activation was used at the final layer to compute the output probabilities. Training was done over a total of 10 epochs with a learning rate of $1e^{-3}$ and a batch size of 64. The model was optimised using the *Adam optimiser* with *cross entropy* as its loss function.

4.4.2 Text

For a black-box model able to predict the sentiment of movie reviews we have used the model proposed by Oostrum (39). Similar to the previous model, convolutional layers were used as hidden layers except that the convolutions were used independently with varying filter sizes to cover the length of the tokenised sentence. Each of these independent, convolutional layers forward their input to a max-pooling layer after which the result was concatenated. Finally, a dropout layer and a fully-connected final layer were used to obtain a final score between 0 and 1 indicating the sentiment of the sentence. The model was trained over 10 epochs with a batch size of 64 with a learning rate of $1e^{-3}$. The model was optimised with the Adam optimiser, using a decaying learning rate with rate $3e^{-4}$. For l

Table 4.2: Performance metrics of the black-box models on their test datasets in terms of predictive accuracy.

Model	**Negative or digit 0**			**Positive or digit 1**		
	precision	recall	F-score	precision	recall	F-score
Binary MNIST model	0.99	0.99	0.99	0.99	0.99	0.99
Movie reviews model	0.74	0.82	0.78	0.85	0.77	0.81

4.5 Explainers

We have used the implementation of our explainers (described in Section 2.2) provided by **DIANNA** (43), an open-source Python package, available on github, providing a unified framework for RISE, LIME and KernelSHAP across a variety of data modalities[1]. DIANNA provides data preprocessing, visualisation and most importantly, supports a standardised deep learning framework: *Open Neural Network eXchange* (**ONNX**). This standardised format for *Deep Learning* (DL) models allows for seamless integration of DL models that have been developed across different frameworks for DL, such as **Tensorflow** and **Pytorch**. Additionally, the aforementioned models and datasets are included in the package as well.

4.5.1 Hyperparameter Configurations

Choosing a relevant set of hyperparameters for XAI methods to investigate their relationship with the explanation quality was not considered trivial, due to both their interactions with explanation quality being unknown and the set of hyperparameters possibly being very large: RISE only has three different hyperparameters, LIME is a subject to about 10 different hyperparameters, depending on the implementation. As such, only several hyperparameters were selected which were deemed to be the most important from our domain

[1] Ranguelova E., Bos P., Meijer, C.W., Liu Y. et al., DIANNA, `https://github.com/dianna-ai/dianna`

knowledge. These appropriate subsets of hyperparameters were grouped together in a grid with all possible combinations of selected hyperparameters. This grid was in turn used to exhaustively compute explanation metrics for each configuration (element) of the grid. For an overview of the chose configurations, we refer to table 4.3. Given the unbounded range of some of our hyperparameters, we have chosen to set some our spaces to logarithmic (multiplicative increase between values) and other to linear (additive increase between values) in order to effectively cover the range of these parameters.

Table 4.3: Hyperparameter configurations per algorithm and modality for the grid search.

Algorithm	Modality	Hyperparameter	Min	Max	Space	Number
RISE	Image	p_{keep}	0.05	0.95	Linear	10
		n_{masks}	400	1900		6
		Resolution	3	19		9
	Text	p_{keep}	0.05	0.95	Linear	20
		n_{masks}	400	1800		10
LIME	Image	$n_{samples}$	400	1600	Linear	4
		$n_{segments}$	20	95	Linear	6
		Kernel Width	5e−2	100	Logarithmic	5
		L2 regularisation	0	3	Logarithmic	5
	Text*	$n_{samples}$	400	1900	Linear	6
		Kernel Width	1	1000	Logarithmic	8
		L2 Regularisation	0	3	Logarithmic	8
KernelSHAP	*Image*	$n_{samples}$	400	1600	Linear	4
		$n_{segments}$	20	95	Linear	6
		L1 regularisation†	0	1e−4	Logarithmic	7

*: Used no feature selection such that all tokens in sentence were given attribution scores.

†: Included the *auto* choice given by the implementation of KernelSHAP.

4.6 Metrics

This section will highlight design choices and implementation details for the evaluation metrics regarding table 4.1.

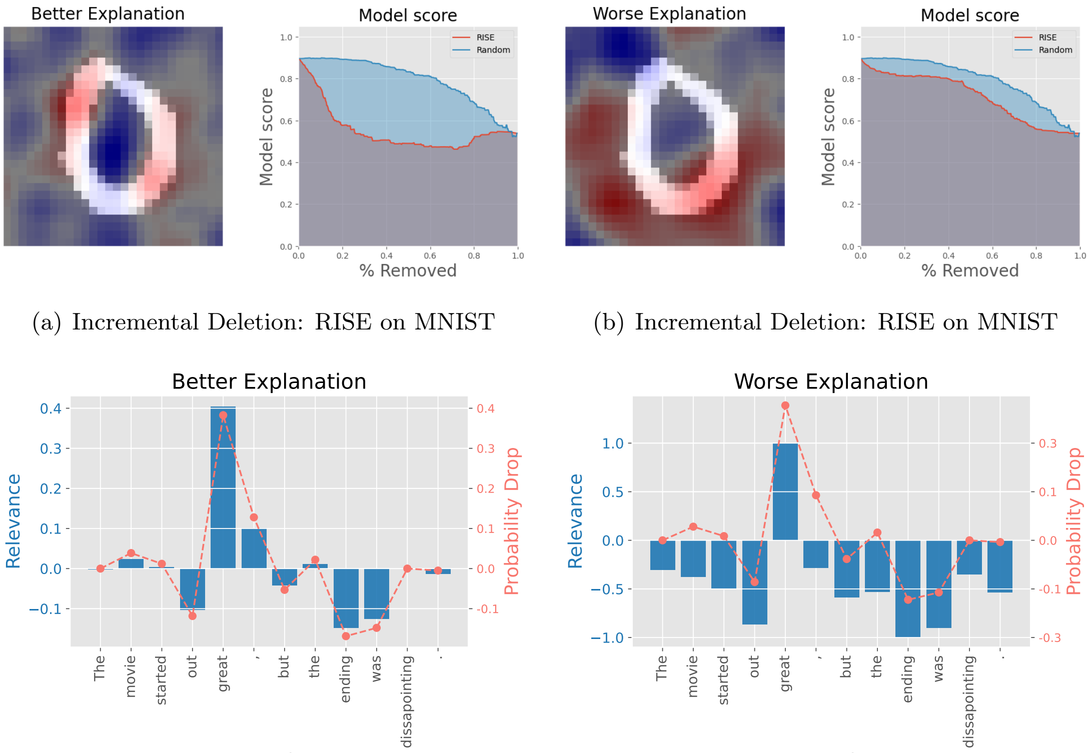
4.6.1 Incremental deletion

Incremental deletion is a metric for evaluating XAI method's correctness. The idea behind incremental deletion is the 'removal' of features in an instance in incremental fashion until all features are removed. Concurrently, the model scores of these perturbed instances are captured to compute metrics, the process is shown on high level in figure 4.2, subfigure a and b. The idea is similar to perturbation-based XAI methods themselves: the removal of features that bear a stronger correlation with the model's output will more drastically effect its scores. This metric has been mentioned in a variety of works under several names (18, 37, 42, 45), this work largely follows the metric proposed by Petsiuk et al. (42). A key difference is the addition of a random removal curve as mentioned in the work of Nauta et al. (37) which can be further used to asses the correctness score when compared to a random baseline. This random baseline serves as a good sanity check for out-of-distribution samples (perturbed samples that no longer correspond to the the distribution of the training data) and helps generalising the metric through showing its relative performance. The area between the random order of removal and the most relevant first order of removal is computed to arrive at a metric.

Parametrisation In our case, we use the mode of the grayscale values to impute pixels with and delete 2 pixels per deletion iteration. We repeat this process 5 times for each hyperparameter configuration to arrive at more robust estimates.

Advantages For image data, incremental deletion is an attractive choice as it requires granular data, or a larger feature space to work with in the first place. The metric also exhibits good scalability through its *step* parameter (how many features to remove per deletion iteration) allowing for very large feature spaces to be processed effectively. Given that this is a ranked metric, it is scale-invariant which avoids the potentially problematic normalisation of feature attributions. Moreover, the metric functions at the level for single features, allowing for an unbiased comparison across XAI algorithms as it does not assume further information about the instance. Finally, integration along the deletion curves allows for a more robust metric. In several cases, the black-box model will behave in a particularly non-deterministic manner. We refer to one such case where the model missed some relevant features in its explanation as *model resurgence* in figure 7.1.

Disadvantages Incremental Deletion might have limited applicability. The metric is dependent on the model score of the original instance, which will vary on both an instance- and model-basis. This does seem to be a disadvantage to metrics for correctness in general, and not just incremental deletion (5). If the model score is particularly low, computing the area between curves might not be effective. Furthermore, in multi-class problems, it remains unclear as to what class the model changes it prediction. Finally, regression problems might require an alternative approach than the area between curves as its final metric and might need some specific tuning to come to a result.



(a) Incremental Deletion: RISE on MNIST

(b) Incremental Deletion: RISE on MNIST

(c) Single Deletion: LIME on movie reviews

(d) Single Deletion: RISE on movie reviews

Figure 4.2: An visual overview of our correctness metrics: *Incremental Deletion* on MNIST in subfigures 4.2(a), 4.2(b) and *Single Deletion* on Stanford movie reviews in subfigures 4.2(c) and 4.2(d). =A poor choice of hyperparameters or XAI algorithm could lead to incorrect explanations which can be measured and visualised using our metrics.

4.6.2 Single deletion

For textual data, XAI evaluation metrics are still relatively underdeveloped, with some of the underlying issues have been highlighted by Jacovi and Goldberg (29). This work presents one of the first approaches to evaluating explanations for textual data. *Single deletion* is a metric for evaluation of XAI correctness which aims to measure to capture the relationship between the model performance with singly removed features and these feature attributions. This metric has been proposed by Jaakkola and Melis (28), where it was applied on tabular data. It is similar to incremental deletion in the sense that the deletion of features will be applied to an input instance while measuring the change in model score at the same time. Instead of deleting all the values in incremental fashion, the deletion is only applied to a single instance at a time. Again, we provide a demonstration in figure 4.2, subfigures c and d.

Parametrisation For our imputation value, we chose *unkown word tokens* or *UNK* which are used in the field of NLP to represent words not present in the training set. Additionally, we compute single deletion for 20 samples per hyperparameter configuration. We chose *Pearson Correlation Coefficient* (PCC) for our correlation metric with a one-sided test for positive values. Pearson correlations with significance values greater than 0.05 were discarded from the results.

Advantages Single Deletion doesn't pose significant perturbations into an instance, which decreases the likelihood of out-of-distribution samples and preservers relevant feature interactions. Moreover, Single Deletion does not require the same level of granularity as Incremental Deletion because it considers a correlation metric instead of integration. Correlation values also tend to be more intuitive to interpret and more generalisble than integrating along the deletion curves.

Disadvantages The core disadvantage of Single Deletion would be the fact that it only looks at the effect of single features on the model. The black-box model most likely looks into interactions of the features as well, which the metric fails to consider. As such, the metric probably does not reflect correctness entirely. Moreover, Single Deletion might not be appropriate for more granular data as this will present problems in terms of scalability and individual features likely not bearing strong relevance in such data, e.g. the effect of a single pixel is likely not relevant towards the prediction of a model.

4.6.3 Sensitivity

Sensitivity is a metric that discerns the continuity of the explanation. The idea behind the sensitivity metric is to measure the change in explanation under small relative input perturbations. The explanation should exhibit robustness for all of these input perturbations, motivated by adversarial attacks in XAI (i.e. manipulation of explanations by adversaries) (14, 19). Specifically, the *Average Sensitivity* and *Max Sensitivity* have been introduced by Yeh et al. (55). *Average sensitivity* aims to capture the mean distance between the original explanation and perturbed explanation given a Monte Carlo sample of these explanations.

We have provided a formal definition in table 4.1. In essence, the *Frobenius norm* of an explanation of the original instance, and an explanation under relatively small input perturbations ϵ is taken and divided by the Frobenius norm of the original explanation. The Frobenius norm is equivalent to the square root of the sum of the squares of the elements of the matrix.

For an implementation of the sensitivity metric we have used **Quantus**, a framework that encompasses a variety of evaluation metrics (24). Quantus allows for a fine-grained control of evaluation metrics and is available on an open-source package on github. [1]. Quantus does not support textual data yet, and we resorted to a modification of their implementation to attune to our needs for textual data. The following paragraphs will list some further design choices with regard to the sensitivity metric.

Advantages The choice of sensitivity is attractive due to its straightforward approach of Monte Carlo sampling, which allows for a relatively robust estimation of the sensitivity metric. The metric does not involve taking any information from the black-box models, as is the case for our correctness metrics, which allows us to generalise beyond the black-box models.

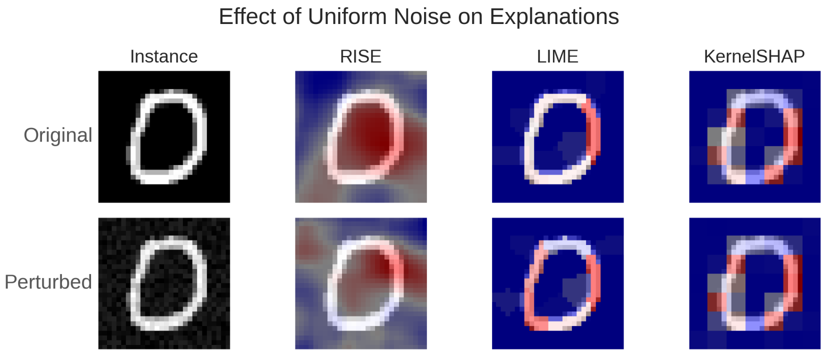
Disadvantages A key disadvantage is that of *scale sensitivity*, where explanations require normalisation before being used in the metric. Normalisation has been stated to possibly harm feature attributions (24). Furthermore, sensitivity depends on the choice of perturbation function and the number of samples used to estimate the metric.

Parametrisation Here we present the parametrisation of the Sensitivity metric, which involves a choice of normalisation functions and perturbation function (ϵ) per modality.

[1] Hedström A. et al. Quantus, `https://github.com/understandable-machine-intelligence-lab/Quantus/`

Normalisation Due to the sensitivity metric being affected by the magnitude of the data, it is required to normalise the explanations before computing the metric. In doing so, a realistic comparison across different XAI algorithms can be made. In line with recent research, we have chosen to normalise around the average second moment estimate that ensures that each attribution score in the relevance map will has an average squared distance to zero that is equal to one (10). We have defined this procedure in equation 4.2, where we denote A as the original attribution map and A_{norm} as its normalised counterpart.

$$A_{norm} = A\sqrt{\frac{n}{\sum_{i=0}^{n} A_i^2}} \tag{4.2}$$



(a)

Original: Uneasy mishmash of styles and genres.

awkward mishmash of styles and genres .

Uneasy mishmash of trend and genres .

Uneasy ragbag of styles and genres .

anxious mishmash of styles and genres .

Uneasy mishmash of styles and genre .

(b)

Figure 4.3: Demonstration of explanation sensitivity through the effect of random input perturbations on attribution maps. In the explanation, red values indicate positive relevance scores, whereas blue values - negative. Uniform noise applied to the original instance- (a) image, (b) text causes the explanations to make perceptually significant changes.

Perturbation function In order to compute the sensitivity metric, an appropriate perturbation function needs to be used (represented as ϵ in the equation for sensitivity in table

4.1).

Images For images, *uniform* or *white noise* was used for perturbation. A visual demonstration of performing these perturbations can be seen in figure 4.3, (a). For parametrisation of the uniform noise, we chose bounds of 0.05 such that the noise would assume values between -0.05 and -0.05, as suggested by the default Quantus parametrisations. Grayscale values for our MNIST dataset ranged between 0 and 1.

Text For textual data (figure 4.3, (b)), finding an appropriate perturbation function proved to be more challenging than for images. In order to assess the robustness of machine learning models for *Natural Language Processing* (NLP), several approaches have been employed by the community in the field of ML itself (17, 38). Some options include human rephrasing of the text or the incorporation of black-box models in order to impute words in the sentence. Instead of these methods we have opted for the more natural approach of replacing a subset of words with their synonyms, similar to the approach by Zhang et al. (56). We have implemented synonym replacement using the thesaurus from *WordNet* (36). We avoided the replacement of stopwords such as 'the', 'I', 'as' etc. Moreover, we considered a uniform distribution and chose to perturb $n_{aug} = \max(\lfloor 0.1 \cdot n_{tokens} \rfloor, 1)$ words in the tokenised text.

4.7 Runtime extraction & Hardware

In the grand scheme of things, most of the algorithmic work is performed on the CPU. Perturbed instances with its model were copied to the GPU to run model inferences, copying back to the CPU for each sample per hyperparameter configuration (θ_i) per instance (x).

All performance measurements are performed on the *Distributed ASCI Supercomputer 5*(DAS-5) (6), using an Intel(R) Xeon(R) CPU E5-2630 v3 CPU. The XPU has two sockets, eight cores per socket and two threads per core, meaning that, theoretically, a total of 32 hardware threads can be used in parallel. The base clock speed of our CPU is 2.40 GHz. In order to run the models, we used an NVIDIA TitanX Maxwell card (also present on DAS-5). The GPU has 3072 *CUDA cores* and runs at a base clock frequency of 1000 MHz.

Runtimes were similarly extracted with the metrics. The median runtime at specific configurations were extracted (5 samples for images and 20 for text). Using Python's `time.time_ns()` function, which leverages the high resolution clock that the system provides (system runs on CentOS Linux 7), we measure the runtime.

5

Results

5.1 Hyperparameter Effects

For the individual hyperparameter effects, we have chosen to aggregate the scores per configuration and individual hyperparameter. The mean for each metric per configuration is taken and consequently the mean for each configuration per hyperparameter value. For sensitivity and incremental deletion we also compute the standard deviation per configuration and the root mean square over the standard deviations across configurations in order to display explanation variance. For Single Deletion, we did not compute deviations, and the median correlation was taken instead of the mean.

5.1.1 RISE

For the effect of individual hyperparameters in RISE several observations can be made. We have outlined our results in figure 5.1. First, p_{keep} was found to be the critical parameter when it comes to explanation continuity. We have found that lower values of p_{keep} typically lead to more sensitive explanations. Interestingly, this pattern was observed *cross-modally*, i.e. true for both images and text. The pattern for text appeared to have especially high sensitivities for particularly low p_{keep}. On the contrary, the pattern for images exhibited a more linear relationship. For the correctness properties, we have found that the values tend to reach an optimum for both p_{keep} as well as resolution. p_{keep} values tended to exhibit optimum values somewhere in the middle ranges, whereas more extreme values exhibited weaker values for both Single deletion and Incremental deletion metrics, especially for the higher p_{keep} values. For incremental deletion, the optimum tended to reside in the lower ranges, when compared to Single Deletion. Furthermore, the Resolution hyperparameter exhibited an optimum at a value of around 6, whilst more extreme values yielded less favourable results.

5.1.2 LIME

In LIME, the hyperparameters effects were similarly measured for individual metrics and summarised in figure 5.2. The most impactful parameter was found to be *Kernel width*. Its effect on explanation sensitivity was most significant, with particularly low values causing a relatively high sensitivity. Inversely, relatively low values of kernel width resulted in higher values of the single deletion metric, whereas the lowest value for kernel width were reflected in lower metric values. These effects appear to converge for larger choices of kernel width. For Incremental deletion, however, no such pattern was found. Interestingly, increasing the number of segments was found to have an adverse effect on explanation quality across both metrics.

5.1.3 KernelSHAP

Finally, the effects for KernelSHAP were only measured for image data (due to the current limitation of the DIANNA library) and the results are presented in figure 5.3. Unsurprisingly, the amount of superpixels was found to be the most influential parameter. Contrary to LIME, KernelSHAP was able to achieve higher scores for incremental deletion when given a larger amount of superpixels or higher $n_{segments}$. Sensitivity, on the contrary, increased for higher values of $n_{segments}$. L1 regularisation was found to have only a marginal detrimental effect on the metrics, with no regularisation, or ordinary least squares regression yielding the best results.

5.2 Cross-Algorithmic Evaluation

In order to compare the different XAI algorithms with each other, we chose to select the best hyperparameter configuration per individual instance x_i and the default choices implemented in DIANNA. Specifically, we computed the rank of Incremental deletion, Single deletion and Sensitivity metrics independently and inverted them (i.e. the lowest score also gets the lowest rank). Next, the *Mean Reciprocal Rank* was computed across metrics to select the best configuration. For each best configuration, the evaluation scores were extracted and their mean was computed. For the Pearson correlations the mean was computed after using the *inverse Fisher* transform. Finally, the runtime of that specific configuration was extracted and similarly aggregated using the mean. We present these results in table 5.1. For the default configurations we refer to table 5.2.

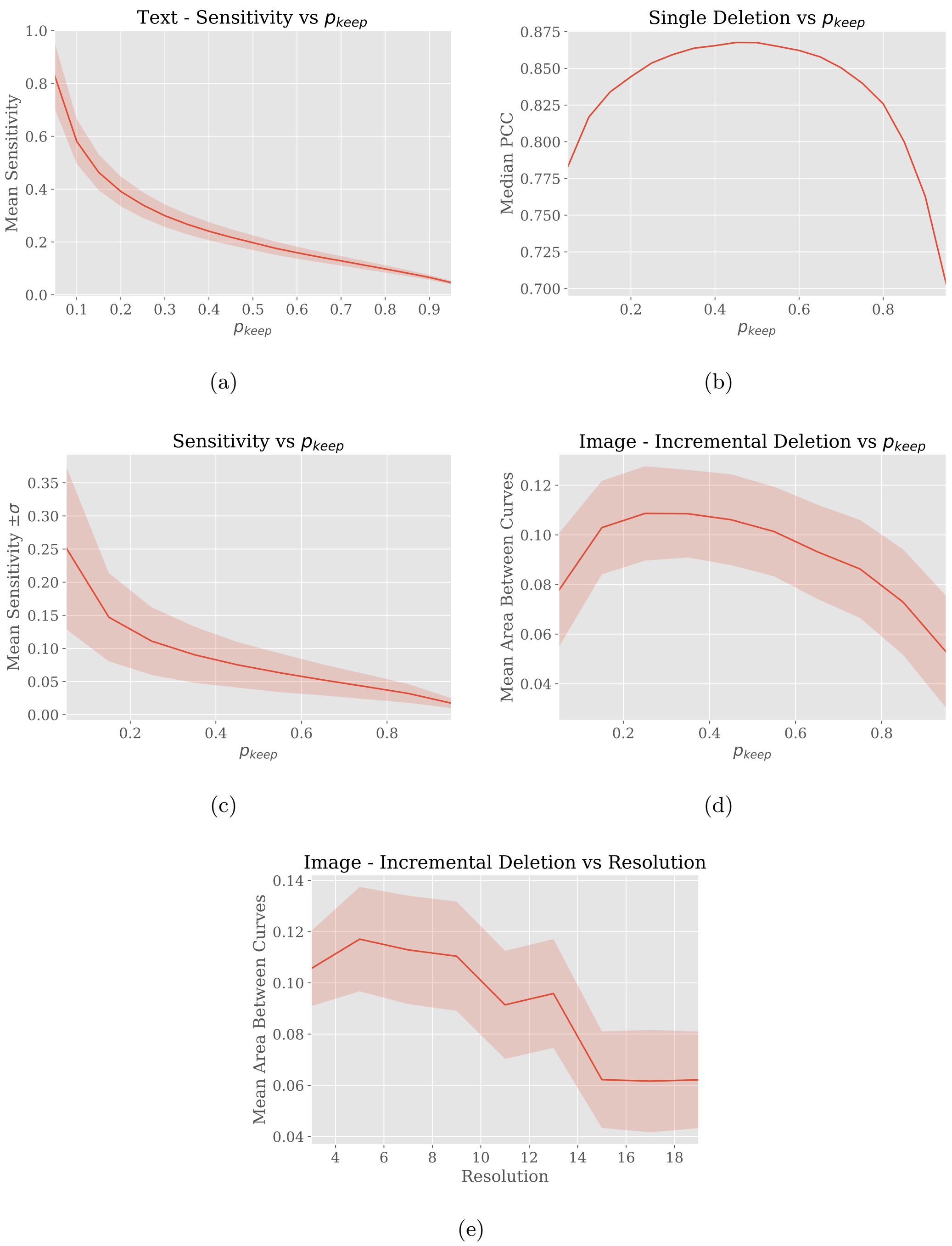
In general, we define the hyperparameter optimisation process as follows:

$$\theta = \underset{\theta_i \in \Theta}{\arg\max} \frac{1}{2} \sum_{i=1}^{2} \frac{1}{\operatorname{rank}(\theta_i)}$$
$$\bar{m} = \underset{x \in X}{\operatorname{agg}}(g_\theta(f, x)),$$

were θ is the optimal hyperparameter choice with θ_i representing a single configuration in the hyperparameter grid. Metrics are than aggregated over the instances x in dataset X with the arithmetic mean or mean in inverse Fisher space and computed back to the probabilistic space:

$$\underset{x \in X}{\operatorname{agg}}(g_\theta(f, x)) = \tanh(\sum_{i=0}^{n} \frac{\operatorname{arctanh}(g_\theta(f, x))}{n})$$

We have observed that KernelSHAP performs better than the other two algorithms in terms of correctness, with RISE coming in second. Its sensitivity is nonetheless higher than for the other two algorithms and its runtime considerably higher. For text metrics, on the other hand, LIME scores higher in terms of Single Deletion, whereas RISE has the better sensitivity score. Interestingly, RISE exhibited lower runtime values due to LIME having some samples with lower values of $n_{samples}$, effectively reducing the amount of work needed to be done by the algorithm. However, when using the default values RISE performs significantly worse due to optimisation in DIANNA. In general, we see that the optimised explanations perform better than the naive ones.



(a)

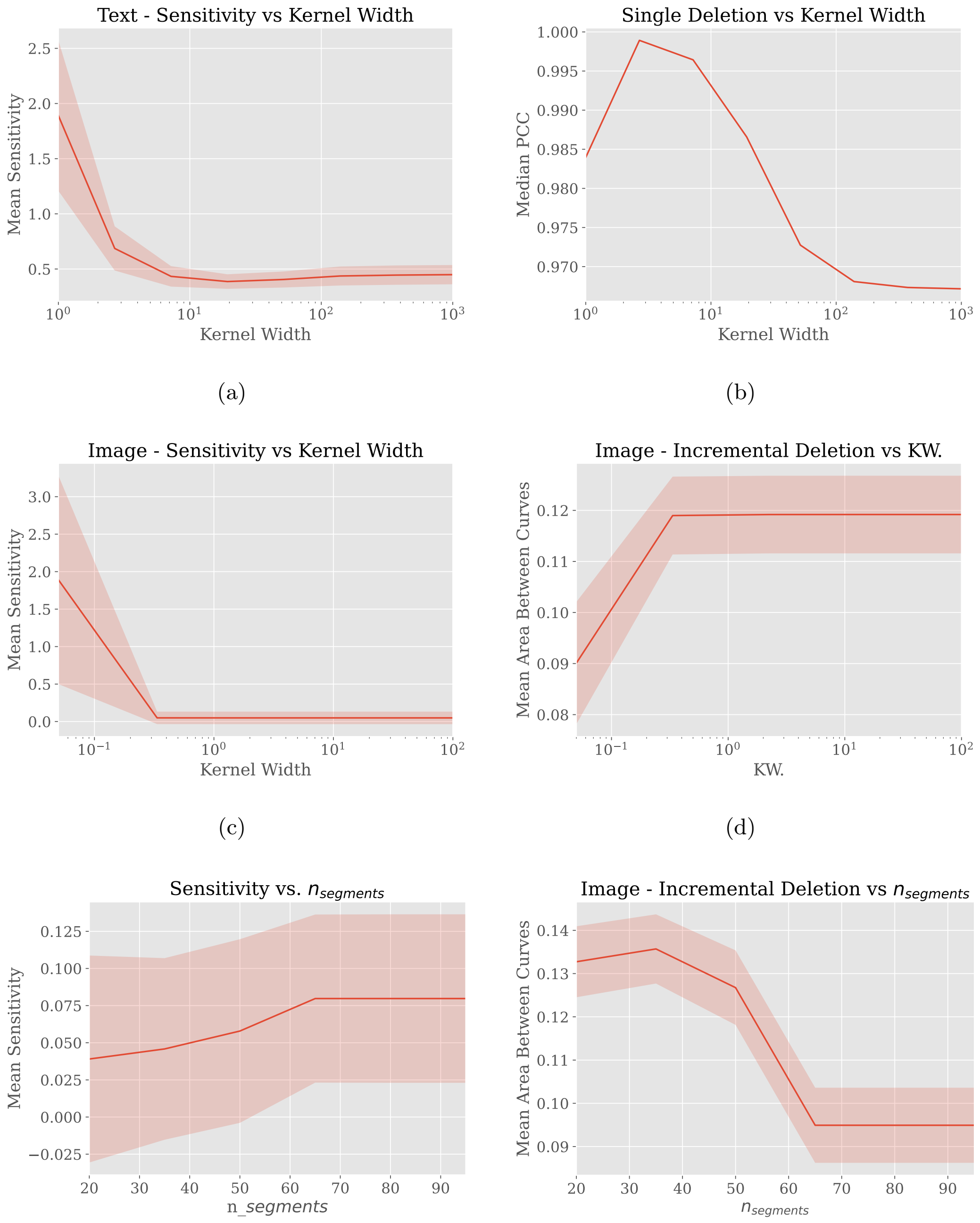
(b)

(c)

(d)

(e)

Figure 5.1: Most relevant individual hyperparameter effects of RISE on XAI evaluation metrics.



(a)

(b)

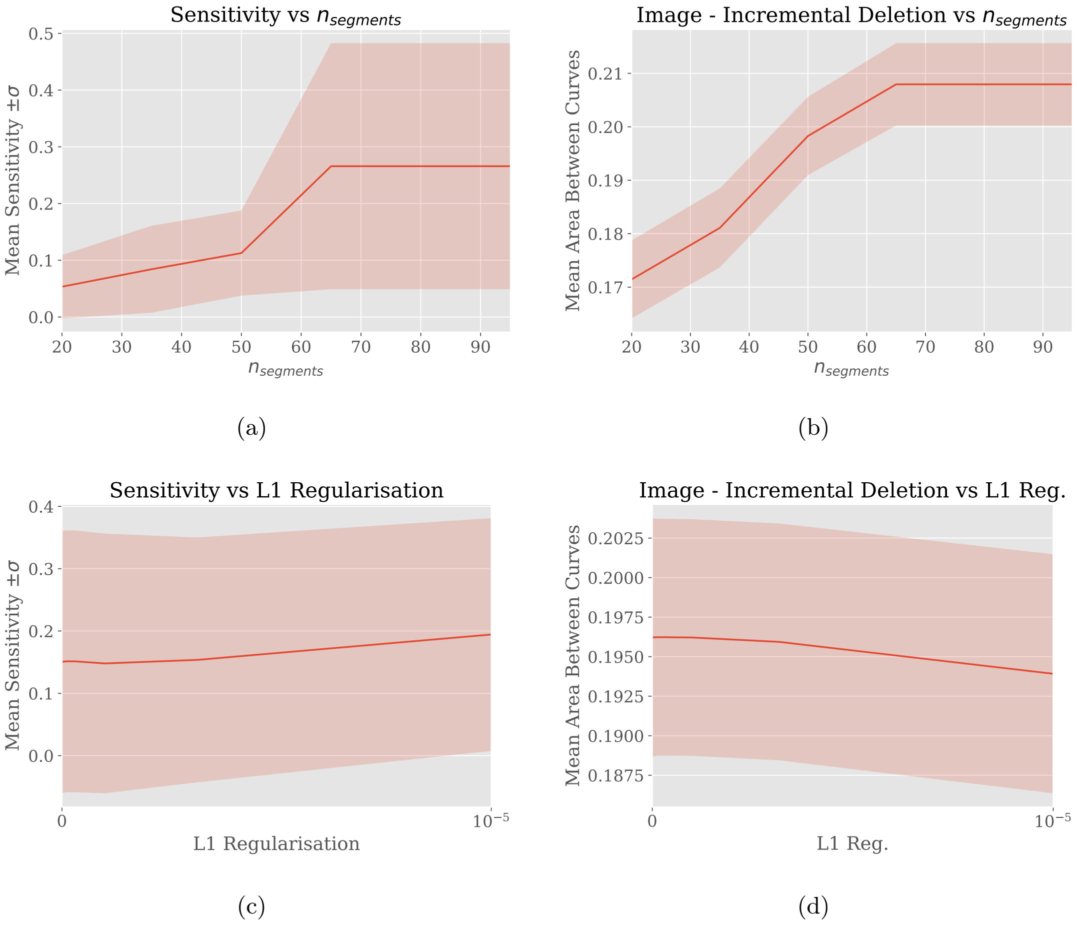
(c)

(d)

(e)

(f)

Figure 5.2: Most relevant Individual hyperparameter effects of LIME on XAI evaluation metrics.



(a)

(b)

(c)

(d)

Figure 5.3: Most relevant Individual hyperparameter effects of KernelSHAP on XAI evaluation metrics.

Table 5.1: Quantitative evaluation metrics of XAI algorithms, using both the optimal values from our ranking strategy and default values which are displayed in table 5.2. ↑ stands for a higher score being desirable, ↓ stands for a lower score being desirable.

Algorithm	**Binary MNIST**			**Stanford Movie Reviews**		
	ID (↑)	SENS (↓)	runtime (ms)	SD (↑)	SENS (↓)	runtime (ms)
RISE (optimal)	0.15	**0.023**	797.74	0.89	0.096	**207.42**
RISE (default)	0.14	0.11	1037.08	0.90	0.29	1662.67
LIME (optimal)	0.12	**0.022**	**624.24**	**0.99**	**0.086**	340.69
LIME (default)	0.10	0.21	648.25	0.98	0.20	412.61
KernelSHAP (optimal)	**0.20**	0.064	2584.43	N.A.		
KernelSHAP (default)	**0.21**	0.19	3955.6			

Algorithm	Hyperparameter	Value(s)
RISE	n_{masks}	1800 (text), 1900 (image)
	p_{keep}	Optimised in DIANNA
	Resolution	6 (Image)
LIME	$n_{samples}$	1800 (text), 1600 (image)
	Kernel Width	25 (Image & Text)
	$n_{segments}$	95
	L2 Regularisation	0.01
KernelSHAP	$n_{samples}$	1600
	$n_{segments}$	95
	L1 Regularisation	auto

Table 5.2: Default hyperparametrisations used for the default option.

6

Discussion

In summary, the goal of this work was multi-fold and consisted of testing several research questions across case studies. The main research question was defined as: How do XAI algorithms and their hyperparametrisations affect explanation quality? Inherent, we formulated the following sub-questions: (1) How can explanation quality be defined qualitatively? (2) How can explanation quality be quantified? (3) What is the relationship between individual hyperparameters and explanation quality? (4) To what extent do explanations benefit from optimisation? and (5) How do explanation methods compare in terms of explanation quality given optimal hyperparameters? The answer of the first two questions are already answered in our approach, hence our interpretation will focus on the latter three.

6.1 Limitations

When considering our approach to this experiment, a large amount of design choices had to be made in order to arrive at the result. We argue that measuring explanation quality relies at least on (1) a selection of representative explanation properties based on audience-dependent desiderata, (2) the definition and implementation of a metric which uses information related to both the black-box model and the explanation to reflect these properties and (3) the extensive process of properly parametrising these metrics. Even though this work considers a thoughtful approach to all three of these constraints, the outcome of this experiment is likely still particularly sensitive to this approach. Hedström et al. (25) provide insights on the latter two points, and introduce the problem of *meta-evaluation* in XAI.

Additionally, we note that our experiments rely on case studies of specific black-box models operating on a limited number of datasets. The image data, in particular, can be considered quite limited as it uses a grayscale coloring scheme and is of relatively small size (28×28 pixels). Our interpretations in the next section 6.2 include an extrapolation of our results to other datasets, which might not be appropriate given the specifics of the used datasets. Additionally, the insights on hyperparameters were obtained using aggregations on many different configurations of other hyperparameters and Monte Carlo samples for each of them. As a result, we obtain more global insights on the XAI hyperparameters; their effect on the dataset as a whole. These general insights might highlight some more general behaviour of the algorithms, but we acknowledge that their effects probably varies strongly based on the underlying data and black-box model.

More specifically, for images in LIME, the feature selection process was left untouched. LIME natively performs feature selection which has likely adversely affected explanation quality.

Another limitation of this work is the usage of grid search for hyperparameter optimisation. Grid search is computationally expensive, just like our XAI metrics, and the combination of the two leads to a particularly compute-intensive endeavour. Given that the goal of this work was to explore the effect of hyperparameters, we found this to be an appropriate approach. For hyperparameter tuning, however, *Bayesian Optimisation*, *Randomized Search* or a *Genetic Algorithm* might be more appropriate (11).

Furthermore, the implementation for the XAI algorithms that was used in this work is not optimised for a GPU, which introduces more scalibility issues. While we run the black-box models on the GPU, some other operations are done on the CPU, which could benefit from hardware acceleration, such as *bilinear interpolation*, *linear regression*, *image masking* etc. Some recent advancements do propose GPU-optimised XAI implementations of Shapley values, promising reasonable increases in performance (41).

6.1.1 Applicability of the sensitivity metric

A phletora of works have been proposed in order to assess explanation continuity in light of relative input perturbations in light of adversarial attacks. On the other hand, after using the quality evaluation metrics, we have found that *explanation consistency* also affects our sensitivity metric due to the inherent randomness of Monte Carlo sampling. The original work by Yeh et al. (55) argue that perturbation-based approaches can have their sensitivity still estimated in this way. Nevertheless, an inflation of both continuity and consistency is introduced by this approach, and it is now unclear which of these properties *sensitivity* reflects. In order to make the distinction between consistency and continuity, we instead argue for fixing the Monte Carlo sampling process to an arbitrary sequence (i.e. using the same random seed for explaining every perturbed instance) and apply the input perturbations to this fixed sequence, whereas consistency can be captured by measuring variability across explanations across samples.

6.2 Interpretation

The effect of individual hyperparameters unveiled several findings across different algorithms and different modalities. The RISE hyperparameter p_{keep} produced an observable pattern in explanation quality where lower values lead to more sensitive explanations, this pattern was observed cross-modally and more specifically, the effect on textual data was even more exacerbated. We expect this exacerbation to be caused by the feature interactions; deleting a relatively high amount of features leads to the deletion of feature interactions, wich affects model volatility. We believe that textual data to bear stronger feature interactions, which explains the more exaggerated relationships between sensitivity and p_{keep}. Furthermore, choosing a lower value for p_{keep} also introduced more imputed instances in the explanation process, which could also lead to more volatile model scores through increased likelihood of out-of-distribution samples. Resolution was also found to strongly influence the incremental deletion metric which reached optimum values between 6 and 8 across the dataset, but varied strongly per instance. We expect the resolution parameter to be indicative of a trade-off between granularity and complexity where lower values reduce the random search space but are less able to capture more granular features. Given that this random search space grows with $O(4^n)$ we still expect lower resolution values to perform better in general.

For LIME, it was found that the kernel width strongly influences explanation consistency, with lower kernel widths causing more unstable explanations. Additionally, for textual data, correctness showed an optimum in the lower regions of kernel width, which possibly hints to a correctness-stability tradeoff for the kernel width parameter. These findings are largely in line with the related work of Visani et al. (54) who described the kernel width as a means to control the locality of the explanation. In opposition, our image data does not reflect this pattern, which might be a result of some unexpected behaviour caused by a combination of the distance metric for the weighting kernel, a gray-scale color scheme and combining these with introduced perturbations. Counterintuitively, introducing more superpixels into the image harmed explanation quality. This coincides with earlier work done by Visani et al. (53), who demonstrated that LIME scales poorly with the dimensionality for the data for tabular data.

In KernelSHAP, the $n_{segments}$ parameter was found to play a pivotal role in explanation quality. The related work of Bansal et al. (7) came to a similar conclusion regarding methods that require segmentation, including LIME. In contrast to LIME, KernelSHAP did scale well with the number of superpixels. Both approaches use linear surrogate models to arrive at their predictions, however a key difference is the weighting kernel. Whereas KernelSHAP relies on game theory to weigh its input perturbations, LIME requires a distance kernel with a specific distance metric to weigh its samples, thus it inherits the drawbacks and biases of this function. In high-dimensional settings such as images, distance metrics are known to exhibit biased behaviour (3).

Finally, across the different algorithms, we found that explanation algorithms vary in terms of their metrics across best settings. XAI algorithms likely behave in a nuanced manner across different data modalities and in terms of explanation properties. In general, we see improved results using optimal hyperparameters.

7

Conclusion

This work explores several unsolved problems in the field of XAI, including an approach to its quantitative evaluation, finding the optimal hyperparameters for and selecting the right XAI algorithm. We advocate for a more holistic view of the concept of "explainability", and argue that it is useful to divide it in several explanation properties according to desiderata of explanation audience. Consequently, in order to measure these properties, we selected and developed several quantitative evaluation metrics in accordance to recent developments in the field. We present our choices and considerations for these metrics along with an extensive and carefully motivated approach to parameterise these metrics. Among these choices we present an approach to quantitative evaluation in XAI for textual data, where this work being one of the first endeavours to do so. Using these metrics, we present a set of case studies on which a grid search was performed in order to evaluate different hyperparameter configurations and their effect on explainability. Several XAI algorithms were explored, including KernelSHAP, LIME and RISE.

7.1 Main findings

Interestingly, some indicative relationships between the XAI hyperparameters and their effect on explanation quality were found, which appear largely consistent with the findings of related work. In RISE, p_{keep} appears to control the continuity of the explanation, but selecting a good value relies on balancing the continuity with correctness. LIME appears especially cumbersome to optimise and finding a good neighbourhood appears especially challenging. KernelSHAP, on the contrary, performed better in terms of faithfulness but requires significantly more time to run, making it less applicable for potential real-time applications. Overall, we find that optimised explanations perform better or equal than the their unoptimised counterparts.

7.2 Future Work

As mentioned in section 6.1 on limitations, using XAI evaluation metrics requires extensive parametrisation, which is a cumbersome process. Furthermore, selecting an appropriate evaluation metric has proven to pose a great challenge as well. Further work on the applicability and performance of metrics could help to improve and give credibility to the results.

Another interesting path to take would be to address one of the key issues in XAI, which is the unwanted generation of out-of-distribution samples through input perturbations. Incorporation of the *RemOve And Retrain* (ROAR) paradigm (26) within the evaluation framework could be a good approach. ROAR aims to mitigate this issue by introducing the perturbed instances to the model and retraining them with it. An interesting endeavour would be to compare the results of performing XAI evaluation on a retrained model and the original model.

Reflecting on our approach for XAI evaluation for text, more work needs to be done in order to gain a grasp on their relevance. Our current approach for assessing correctness, Single Deletion, stresses the importance of removing single words out of context which may not look into all of the black-box behaviour. For correctness, exploring metrics that perturb multiple words as a subset of the entire sentence, like the one proposed by Bhatt et al. (9) would be interesting to investigate and compare to the current approach. Moreover, our approach to input perturbation for text utilises synonym replacement in a simple straightforward manner. Nonetheless, it is definitely worth to investigate alternatives.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps, 2020. 2

[2] Chirag Agarwal, Satyapriya Krishna, Eshika Saxena, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. Openxai: Towards a transparent evaluation of model explanations, 2023. 2, 9

[3] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the 8th International Conference on Database Theory*, ICDT '01, page 420–434, Berlin, Heidelberg, 2001. Springer-Verlag. ISBN 3540414568. 48

[4] Christopher J. Anders, Plamen Pasliev, Ann-Kathrin Dombrowski, Klaus-Robert Muller, and Pan Kessel. Fairwashing explanations with off-manifold detergent. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020. 8

[5] Leila Arras, Ahmed Osman, and Wojciech Samek. Clevr-xai: A benchmark dataset for the ground truth evaluation of neural network explanations. *Inf. Fusion*, 81(C): 14–40, may 2022. ISSN 1566-2535. doi: 10.1016/j.inffus.2021.11.008. URL `https://doi.org/10.1016/j.inffus.2021.11.008`. 21, 29

[6] Henri Bal, Dick Epema, Cees de Laat, Rob van Nieuwpoort, John Romein, Frank Seinstra, Cees Snoek, and Harry Wijshoff. A medium-scale distributed system for computer science research: Infrastructure for the long term. *Computer*, 49(5):54–63, 2016. doi: 10.1109/MC.2016.127. 34

[7] Naman Bansal, Chirag Agarwal, and Anh Nguyen. Sam: The sensitivity of attribution methods to hyperparameters. pages 8670–8680, 06 2020. doi: 10.1109/CVPR42600.2020.00870. 20, 48

[8] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2019.12.012. URL https://www.sciencedirect.com/science/article/pii/S1566253519308103. 15, 17

[9] Umang Bhatt, Adrian Weller, and José M. F. Moura. Evaluating and aggregating feature-based model explanations. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3016–3022. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/417. URL https://doi.org/10.24963/ijcai.2020/417. Main track. 50

[10] Alexander Binder, Leander Weber, Sebastian Lapuschkin, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Shortcomings of top-down randomization-based sanity checks for evaluations of deep neural network explanations, 2022. 33

[11] Robin Cugny, Julien Aligon, Max Chevalier, Geoffrey Roman Jimenez, and Olivier Teste. Autoxai. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. ACM, oct 2022. doi: 10.1145/3511808.3557247. 3, 20, 46

[12] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 25

[13] Abhirup Dikshit and Biswajeet Pradhan. Interpretable and explainable ai (xai) model for spatial drought prediction. *Science of The Total Environment*, 801:149797, 2021. ISSN 0048-9697. doi: https://doi.org/10.1016/j.scitotenv.2021.149797. URL https://www.sciencedirect.com/science/article/pii/S0048969721048725. 1

[14] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.,

2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bb836c01cdc9120a9c984c525e4b1a4a-Paper.pdf. 2, 31

[15] Finale Doshi-Velez and Been Kim. Considerations for evaluation and generalization in interpretable machine learning. 2018. URL https://api.semanticscholar.org/CorpusID:52840192. 17

[16] Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science Advances*, 4(1):eaao5580, 2018. doi: 10.1126/sciadv.aao5580. URL https://www.science.org/doi/abs/10.1126/sciadv.aao5580. 1

[17] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.84. URL https://aclanthology.org/2021.findings-acl.84. 34

[18] Ruth Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *CoRR*, abs/1704.03296, 2017. URL http://arxiv.org/abs/1704.03296. 29

[19] Amirata Ghorbani, Abubakar Abid, and James Zou. INTERPRETATION OF NEURAL NETWORK IS FRAGILE, 2018. URL https://openreview.net/forum?id=H1xJjlbAZ. 2, 31

[20] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, 2018. doi: 10.1109/DSAA.2018.00018. 17

[21] David Gunning and David Aha. Darpa's explainable artificial intelligence (xai) program. *AI Magazine*, 40(2):44–58, Jun. 2019. doi: 10.1609/aimag.v40i2.2850. URL https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2850. v, 2

[22] Hajar Hakkoum, Ibtissam Abnane, and Ali Idri. Interpretability in the medical field: A systematic mapping and review study. *Applied soft computing*, 117:108391–, 2022. ISSN 1568-4946. 1

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society. doi: 10.1109/CVPR.2016.90. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.90`. vi, 60

[24] Anna Hedström, Leander Weber, Daniel Krakowczyk, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina Marina M.-C. Höhne. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research*, 24(34):1–11, 2023. URL `http://jmlr.org/papers/v24/22-0142.html`. 2, 5, 32

[25] Anna Hedström, Philine Bommer, Kristoffer K. Wickstrøm, Wojciech Samek, Sebastian Lapuschkin, and Marina M. C. Höhne. The meta-evaluation problem in explainable ai: Identifying reliable estimators with metaquantus, 2023. v, 17, 45

[26] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. *A Benchmark for Interpretability Methods in Deep Neural Networks*. Curran Associates Inc., Red Hook, NY, USA, 2019. 50

[27] Mir Riyanul Islam, Mobyen Uddin Ahmed, Shaibal Barua, and Shahina Begum. A systematic review of explainable artificial intelligence in terms of different application domains and tasks. *Applied Sciences*, 12(3), 2022. ISSN 2076-3417. doi: 10.3390/app12031353. URL `https://www.mdpi.com/2076-3417/12/3/1353`. 1

[28] Tommi Jaakkola and David Melis. Towards robust interpretability with self-explaining neural networks. 01 2018. 24, 30

[29] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Annual Meeting of the Association for Computational Linguistics*, 2020. URL `https://api.semanticscholar.org/CorpusID:215416110`. 30

[30] Markus Langer, Daniel Oster, Timo Speith, Holger Hermanns, Lena Kästner, Eva Schmidt, Andreas Sesing, and Kevin Baum. What do we want from explainable artificial intelligence (xai)? – a stakeholder perspective on xai and a conceptual model guiding interdisciplinary xai research. *Artificial Intelligence*, 296:103473, 2021. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2021.103473. URL `https://www.sciencedirect.com/science/article/pii/S0004370221000242`. 17

[31] Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10:e0130140, 07 2015. doi: 10.1371/journal.pone.0130140. 21

[32] Stan Lipovetsky and Michael Conklin. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17:319 – 330, 10 2001. doi: 10.1002/asmb.446. 14

[33] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. 12 2017. 14

[34] Harsh Mankodiya, Mohammad S. Obaidat, Rajesh Gupta, and Sudeep Tanwar. Xai-av: Explainable artificial intelligence for trust management in autonomous vehicles. In *2021 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, pages 1–5, 2021. doi: 10.1109/CCCI52664.2021.9583190. 1

[35] Christiaan Meijer and Yang Liu. Onnx model trained on the binary mnist dataset, January 2022. URL `https://doi.org/10.5281/zenodo.5907177`. 26

[36] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38 (11):39–41, nov 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL `https://doi.org/10.1145/219717.219748`. 34

[37] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Comput. Surv.*, feb 2023. ISSN 0360-0300. doi: 10.1145/3583558. URL `https://doi.org/10.1145/3583558`. Just Accepted. vii, 2, 15, 16, 19, 29

[38] Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. Evaluating robustness to input perturbations for neural machine translation. In *ACL 2020*, 2020. URL `https://www.amazon.science/publications/evaluating-robustness-to-input-perturbations-for-neural-machine-translation`. 34

[39] Leon Oostrum. ONNX and PyTorch models trained on Stanford sentiment treebank dataset, January 2022. URL `https://doi.org/10.5281/zenodo.5910598`. 26

[40] Frederik Pahde, Galip Ümit Yolcu, Alexander Binder, Wojciech Samek, and Sebastian Lapuschkin. Optimizing explanations by network canonization and hyperparameter search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3818–3827, June 2023. 21

[41] Zhixin Pan and Prabhat Mishra. Hardware acceleration of explainable machine learning. In *Proceedings of the 2022 Conference & Exhibition on Design, Automation & Test in Europe*, DATE '22, page 1127–1130, Leuven, BEL, 2022. European Design and Automation Association. ISBN 9783981926361. 46

[42] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018. v, 10, 11, 24, 29

[43] Elena Ranguelova, Christiaan Meijer, Leon Oostrum, Yang Liu, Patrick Bos, Giulia Crocioni, Matthieu Laneuville, Bryan Cardenas Guevara, Rena Bakhshi, and Damian Podareanu. Dianna: Deep insight and neural network analysis. *Journal of Open Source Software*, 7(80):4493, 2022. doi: 10.21105/joss.04493. URL `https://doi.org/10.21105/joss.04493`. 5, 27

[44] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016. 12

[45] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11): 2660–2673, 2017. doi: 10.1109/TNNLS.2016.2599820. 24, 25, 29

[46] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. doi: 10.1109/ICCV.2017.74. 9

[47] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `https://aclanthology.org/D13-1170`. 25

[48] Kacper Sokol and Peter Flach. Explainability fact sheets: A framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 56–67, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369367. doi: 10.1145/3351095.3372870. URL `https://doi.org/10.1145/3351095.3372870`. 15

[49] Timo Speith. A review of taxonomies of explainable artificial intelligence (xai) methods. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 2239–2250, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3534639. URL `https://doi.org/10.1145/3531146.3534639`. v, 7, 8

[50] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org, 2017. 9

[51] Tom Vermeire, Thibault Laugel, Xavier Renard, David Martens, and Marcin Detyniecki. How to choose an explainability method? towards a methodical implementation of xai in practice, 2021. 3

[52] Giulia Vilone and Luca Longo. Explainable artificial intelligence: a systematic review, 2020. 9

[53] Giorgio Visani, Enrico Bagli, Federico Chesani, Alessandro Poluzzi, and Davide Capuzzo. Statistical stability indices for lime: Obtaining reliable explanations for machine learning models. *Journal of the Operational Research Society*, 73:1–11, 02 2021. doi: 10.1080/01605682.2020.1865846. 48

[54] Giorgio Visani, Enrico Bagli, and Federico Chesani. Optilime: Optimized lime explanations for diagnostic computer algorithms, 2022. 20, 47

[55] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. *On the (in)Fidelity and Sensitivity of Explanations.* Curran Associates Inc., Red Hook, NY, USA, 2019. 24, 31, 46

[56] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf`. 34

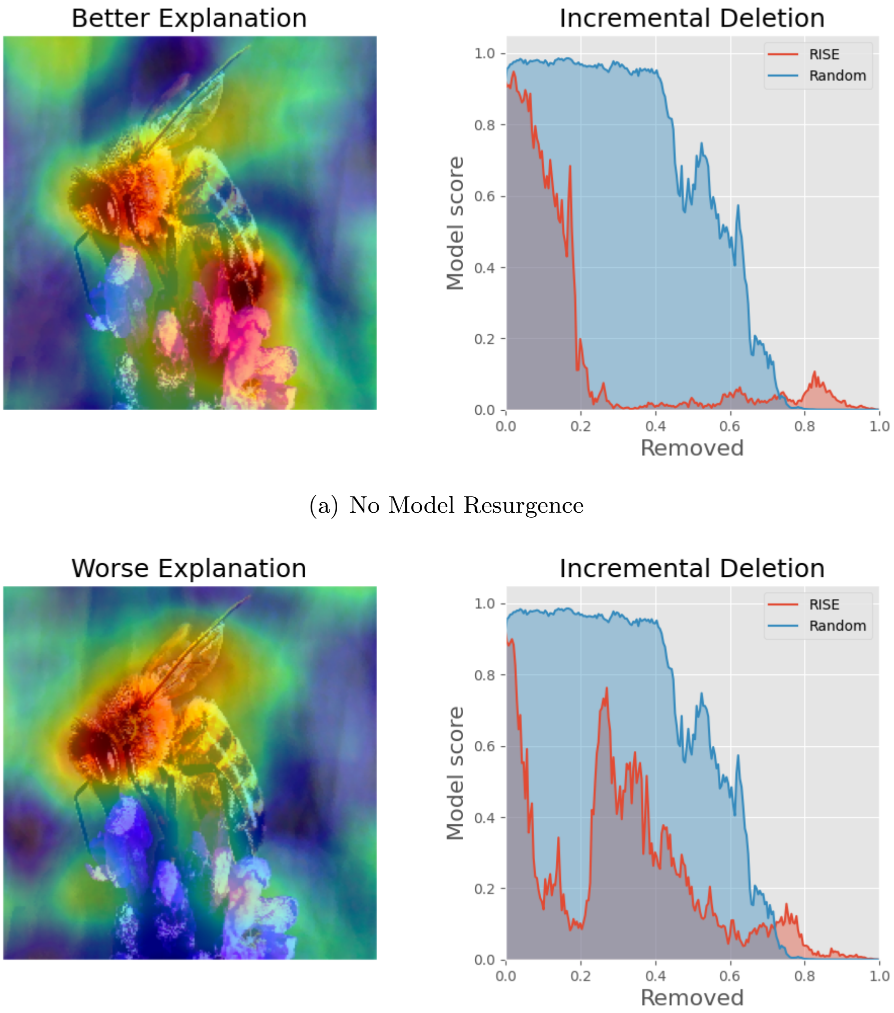
Appendix

7.3 Background

Table 7.1: Hyperparameter cper algorithm and modality with a short description.

Algorithm	**Hyperparam**	**Description**
RISE	n_{masks}	The number of masks to use, n in equation 2.1.
	p_{keep}	Probability to keep a value in the seed mask, $\mathbb{E}[M]$ in equation 2.1.
	Resolution	Size of the seed mask, which is square.
LIME	$n_{samples}$	Number of perturbations to use, z' in equation 2.3.
	Kernel Width	A size parameter for the distance kernel, x in equations 2.2 and 2.3.
	L2 Regularisation	A parameter specific to ridge regression, the linear model LIME uses, reflects the $\Omega(g)$ term in equation 2.2.
	$n_{segments}$	The number of superpixels to use, obtained through some segmentation algorithm.
KernelSHAP	$n_{samples}$	Number of perturbations to use, z' in equation 2.4
	L1 Regularisation	Regularisation for LASSO, the linear model that KernelSHAP uses, again refers to complexity term $\Omega(g)$.
	$n_{segments}$	Similar to LIME, the amount of superpixels to useobtained through some segmentation algorithm.

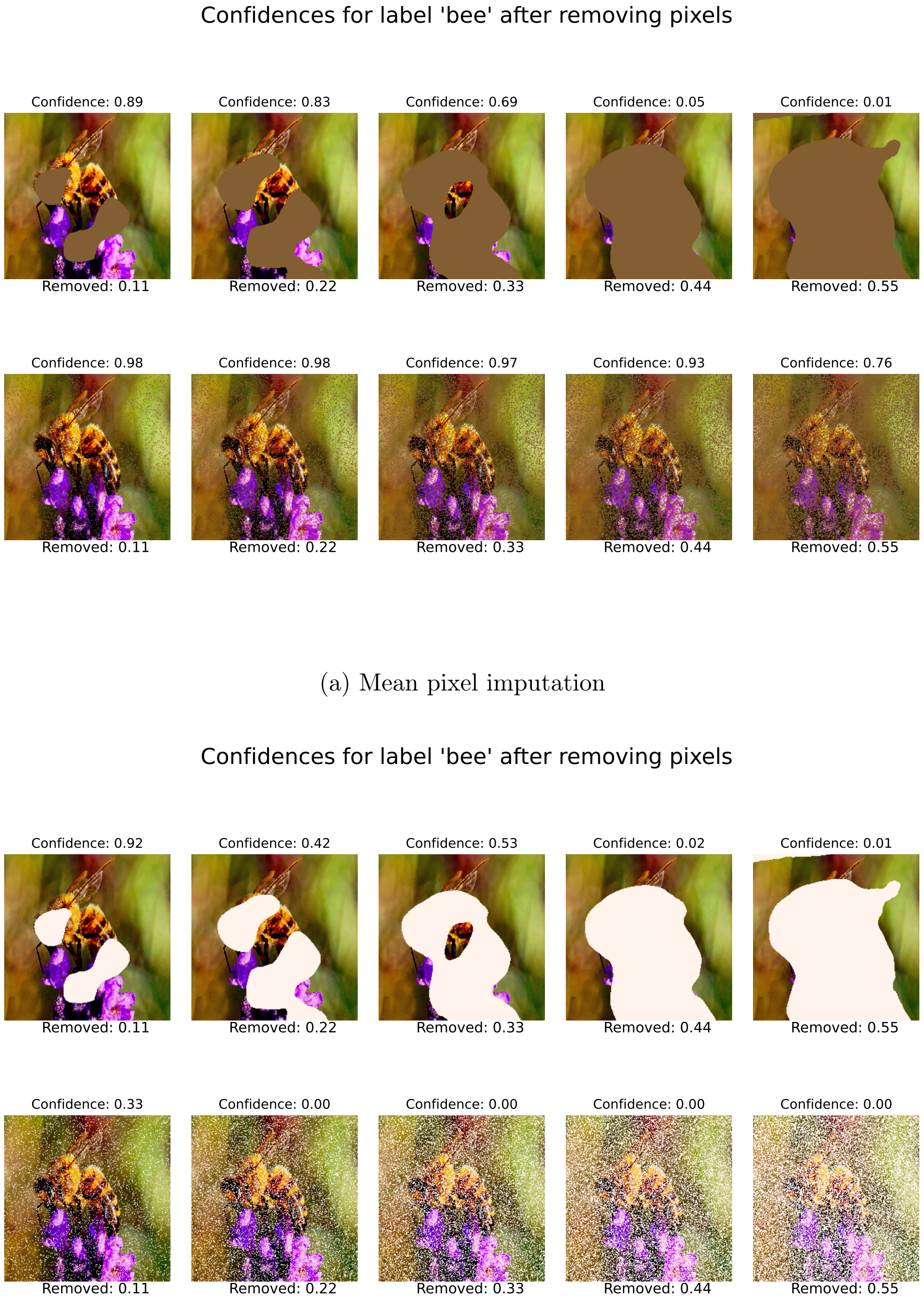
7.4 Experimental Design



(a) No Model Resurgence

(b) Model Resurgence

Figure 7.1: A demonstration of *Model Resurgence* in Incremental Deletion. Using *ResNET* (23) , the model arrives at prediction *bee*. We theorise that when the model misses some relevant features (the stinger of the bee in this case), a model could again arrive at its original explanation. This behaviour is indicative of poor *output-completeness* (see table 2.1).



(a) Mean pixel imputation

(b) White pixel imputation

Figure 7.2: A demonstration of the perils of choosing an imputation value for *Incremental Deletion*. Choosing arbitrary values (white pixels) could lead to *out-of-distribution* samples, causing the model to improperly score the image. To an extent, the random baseline helps in catching such errors.