Warsaw University of Technology

FACULTY OF
MATHEMATICS AND INFORMATION SCIENCE

# Master's diploma thesis

in the field of study Data Science

## From manipulating to evaluating explanations of machine learning models

Hubert Baniecki

student record book number 290761

thesis supervisor
dr hab. inż. Przemysław Biecek, prof. uczelni

WARSAW 2022

## Abstract

From manipulating to evaluating explanations of machine learning models

The trust in learning algorithms is largely correlated with the extent to which we can understand a predictive model. (Un)fortunately, a relatively new line of research in adversarial and responsible machine learning provides various ways of manipulating explanations by altering either model or data. These effectively highlight flaws and vulnerabilities of the existing methods, increasing the ever-growing need to evaluate explanations. In this work, we introduce a unified approach to manipulating model explanations, which is a highly-versatile genetic algorithm targeting many explanations of any black-box model. We use it to perform a large-scale benchmark incorporating typical model analysis scenarios to quantitatively evaluate the robustness of explanations. The results put explainability in the context of the well-known classical machine learning challenges like the curse of dimensionality and bias-variance tradeoff. We propose a local and a global method of evaluating explanations to guide stakeholders in interpreting their results. Experiments on both synthetic and medical data show that explanations of neural networks are far more vulnerable to data poisoning than those of gradient boosting decision trees.

**Keywords:** explainable AI, adversarial machine learning, attacks, resistance

## Streszczenie

Od manipulacji do ewaluacji wyjaśnień modeli uczenia maszynowego

Zaufanie do algorytmów uczących się jest w dużej mierze skorelowane z tym, w jakim stopniu jesteśmy w stanie zrozumieć model predykcyjny. (Nie)stety, stosunkowo nowe badania nad adwersaryjnym i odpowiedzialnym uczeniem maszynowym dostarczają różnych sposobów manipulowania wyjaśnieniami poprzez zmianę modelu lub danych. Skutecznie uwypuklają one wady i słabości istniejących metod, zwiększając stale rosnącą potrzebę ewaluacji wyjaśnień. W niniejszej pracy przedstawiamy ujednolicone podejście do manipulacji wyjaśnieniami modelu, które opiera się na wysoce uniwersalnym algorytmie genetycznym wycelowanym w wiele wyjaśnień dowolnego modelu czarnej skrzynki. Wykorzystujemy go do przeprowadzenia obszernych testów obejmujących typowe scenariusze analizy modeli, aby ilościowo zbadać odporność wyjaśnień. Otrzymane wyniki omawiają wyjaśnialność w kontekście dobrze znanych klasycznych wyzwań uczenia maszynowego, takich jak klątwa wymiarowości i kompromis między obciążeniem a wariancją. Proponujemy metody lokalną i globalną oceny wyjaśnień, aby pomóc interesariuszom w interpretacji ich wyników. Eksperymenty na danych syntetycznych i medycznych pokazują, że wyjaśnienia sieci neuronowych są znacznie bardziej podatne na zatrucie danych niż wyjaśnienia wzmocnionych drzew decyzyjnych.

**Słowa kluczowe:** wyjaśnialna SI, adwersaryjne uczenie maszynowe, ataki, odporność

# Contents

# 1. Introduction

Explainable and interpretable machine learning has become a mature branch of artificial intelligence (AI) research with impactful novel methods and reliable software solutions (Ribeiro et al., 2016; Lundberg & Lee, 2017; Molnar et al., 2018; Samek et al., 2019; Barredo-Arrieta et al., 2020; Biecek & Burzykowski, 2021). Consequently, a successful application of post-hoc model explanations and deep interpretable models allowed for novel developments across various domains like healthcare and social sciences (Samek et al., 2019; Barredo-Arrieta et al., 2020). The available methodology has changed drastically over the last decades, which lead to the wide adoption of an algorithmic modeling culture over the data-focused approach (Breiman, 2001). Explainability enables utilizing black-boxes for knowledge discovery through global explanations and high-stakes decision making through local ones. However, recent criticism suggests that we still know little about the *evaluation* of these methods, especially concerning the potential end-user (Vilone & Longo, 2021). In many cases, the effect of a wrong interpretation is detrimental, and a careless adoption of explanations becomes irresponsible (Rudin, 2019). In fact, we witness an analogous discussion concerning model bias and fairness in machine learning (Corbett-Davies & Goel, 2018; Mehrabi et al., 2021a). It is carried by an increasing number of approaches aiming to *manipulate* and fool model explanations and fairness metrics (Ghorbani et al., 2019; Aivodji et al., 2019; Dombrowski et al., 2019; Heo et al., 2019; Dimanov et al., 2020; Slack et al., 2020; Fukuchi et al., 2020). We acknowledge that these resemble well-known concepts introduced previously in the domain of adversarial machine learning (N. Liu et al., 2021; Sa. Mishra et al., 2021).

Can we *trust* model explanations? In general, these are developed to best reflect the model's reasoning, e.g. allow for a meaningful interpretation of its predictions (Ribeiro et al., 2016). More specifically, we should always consider explanations in a broader context associated with a given scenario and data. A responsible approach to machine learning assumes accounting for model bias, human perception, and crucially for this work, security and safety (Gill et al., 2020). We can address some of the trust issues concerning explanations by using comprehensive machine learning tools for interactive model analysis and monitoring (Baniecki et al., 2021). Although it might suffice for machine learning developers and researchers, which are the early adopters of the aforementioned methods, the number and variety of stakeholders in the area of explainable AI increase steadily (Barredo-Arrieta et al., 2020). Therefore, new methods for quantifying trust and the overall context of explanations become urgently needed (Chatila et al., 2021).

We focus on the application of explanatory model analysis in biology, medicine, and healthcare due to its relatively early and extensive adoption (Holzinger et al., 2019; Lundberg et al., 2020; Markus et al., 2021). It is crucial to distinguish the rather technical explainability framework from *causability*, where the latter can be defined as *"the extent to which an explanation of a statement to a human expert achieves a specified level of causal understanding with effectiveness, efficiency and satisfaction in a specified context of use"* (Holzinger et al., 2019). Manipulating explanations alters its interpretation; thus, it lowers the efficiency and causability capabilities.

**Motivation.** This work originates from the prototype method of manipulating explanations presented by us in *"Manipulating SHAP via Adversarial Data Perturbations"* (Baniecki & Biecek, 2022), where we have shown the following adversarial machine learning scenario from the healthcare domain. There is a black-box achieving good performance in predicting the presence of heart disease in a patient.[1] It predicts a borderline probability of about 0.5 for a 46-year old female patient characterised by clinical variables reported in the axis of Figure 1.1. Especially in such ambiguous cases, medical stakeholders require an explanation of the prediction for the model to be helpful in practice. Therefore, the blue bars in Figure 1.1 present attributions of each variables' values into this exact prediction, which can be estimated for example with SHAP (Lundberg & Lee, 2017). At first, we acknowledge the impact of the variables `oldpeak` and `sex` as locally most important to the model. Conventionally, one would incorporate protected variables like `sex` into the model to control their significance, which at times might discredit the model's prediction by highlighting its bias coming from the data. In this case, either the model developer or auditor might want to manipulate such an explanation to visually lower the protected variable's attribution, effectively providing false evidence of insignificance. In Figure 1.1, the grey bars show an arbitrary target explanation that one wants to achieve by attacking either model or data in the machine learning pipeline, while the obtained manipulated explanation for this scenario is in red. An adversarial attack leads to a change in interpretation: an impact of `oldpeak` and `trestbps` is locally most important, while `sex` and `age` attribute the lowest. Overall, even a seemingly slight possibility of manipulating explanations results in trust issues concerning the applicability of explainable machine learning and black-boxes in general. We cannot consider the causability and understanding of a human expert without accounting for the points of failure in our systems; thus, evaluating explanations in the context of an adversary becomes mandatory. The above conclusion from Baniecki & Biecek (2022) motivates us to perform an extensive benchmark of explanations in various manipulation scenarios.

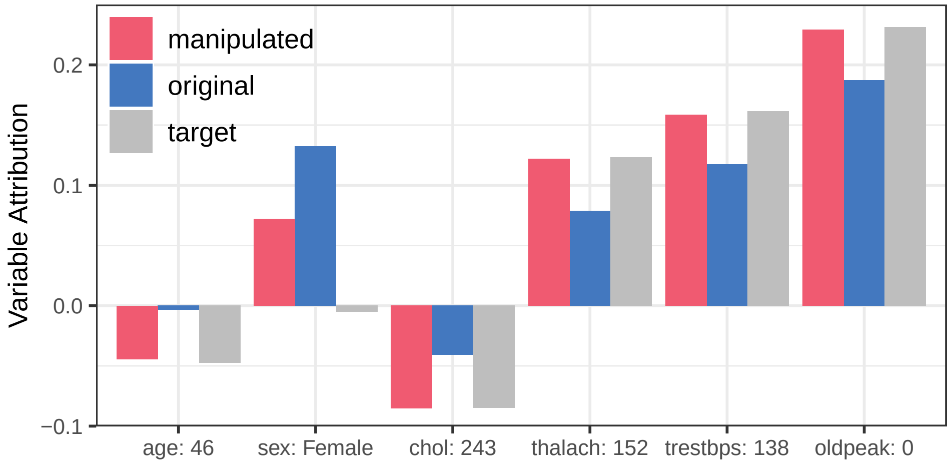

Figure 1.1: Original and manipulated local explanations of a prediction in the `heart` disease classification task (Baniecki & Biecek, 2022). The adversarial attack aims for an arbitrarily set target (in grey). Attribution of the `sex` variable diminishes as other become more prevalent.

[1] In this example, we revisit the well-known Heart Disease dataset available in the UCI Machine Learning Repository at `https://archive.ics.uci.edu/ml/datasets/heart+disease`.

**Contribution.** The contribution of this work is as follows:

1. We survey recent advancements in adversarial and explainable machine learning with a particular emphasis on providing a useful taxonomy of methods and discussing their results presented at top machine learning conferences over the last three years.

2. We introduce and implement a highly-versatile genetic algorithm for manipulating many explanations of any black-box model, which is an extension of our previous work (Baniecki & Biecek, 2022).

3. We perform a large-scale benchmark of manipulating explanations in various scenarios to quantitatively evaluate their robustness and vulnerabilities. This is to put explainable machine learning in the context of classical learning challenges like the curse of dimensionality and bias-variance tradeoff.

# 2. Related Work

The following sections provide the background related to our work, specifically considering adversary and evaluation in explainable machine learning. We refer the reader to the appropriate survey articles for an introduction to this otherwise broad domain (Barredo-Arrieta et al., 2020; N. Liu et al., 2021; Vilone & Longo, 2021; Minh et al., 2021). Sa. Mishra et al. (2021) is the first work to attempt surveying methods concerning the robustness of explanations, yet in a concise manner, which we aim to extend here.

## 2.1. SHapley Additive exPlanations

We explicitly target SHapley Additive exPlanations (SHAP) (Lundberg & Lee, 2017) in our study, which is the most popular, and widely considered state-of-the-art (Holzinger et al., 2022), approach for interpreting any machine learning prediction. It originates from game theory where Shapley (1953) introduced values for n-person games. In the context of explainability, these can be used to formalize the attribution of variables into the model's outcome.

**Definition 2.1 (Shapley values (Shapley, 1953)).** The Shapley value $\phi$ of a variable $j$ for a model $f$ predicting an outcome for an observation $x^*$ is defined as

$$\phi_j(f,\, x^*) = \sum_{S \subseteq V \setminus \{j\}} \frac{|S|!(|V| - |S| - 1)!}{|V|!} [f_{S \cup \{j\}}(x_{S \cup \{j\}}) - f_S(x_S)], \tag{2.1}$$

where $S$ is a subset of all variables $V$, $|S|$ denotes its size, $x_S$ represents the input values of variables in $S$, and $f_S(x_S)$ becomes an expected value of function $f$ conditional on the $x^*_S$ variables' values, meaning $f_S(x_S) = \mathbb{E}[f(x)|x_S = x^*_S]$.

Štrumbelj & Kononenko (2010, 2014) first introduced quasi-random and adaptive sampling algorithms that allowed approximating the Shapley values for black-box models; otherwise computed in exponential time due to taking into account all subsets $S \subseteq V \setminus \{j\}$. Lundberg & Lee (2017) introduced a class of *additive variable attribution methods*, which unified several existing explanations, and proposed SHAP value as an alternative approximation solution to Equation 2.1. SHAP values became broadly adopted in machine learning applications after the introduction of a highly efficient TreeSHAP algorithm (Lundberg et al., 2020) that allowed computing explanations for tree-ensembles in polynomial time. This became a breakthrough because gradient boosting decision trees became the state-of-the-art algorithm for constructing machine learning predictive models. TreeSHAP allowed computing SHAP values for a vast number of samples at once, which brings extended insights into a black-box by visualizing global explanations like variable importance, dependence, and interactions (Lundberg et al., 2020).

Worth mentioning are baseline model-agnostic methods that are often compared against SHAP like Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016) and Break-down (Staniak & Biecek, 2018) against local SHAP attributions, and Permutation-based Variable Importance (PVI) (Fisher et al., 2019) against global SHAP importance. These can be evaluated in several dimensions, which we further discuss in Section 2.3. LIME relies on fitting an interpretable surrogate model, e.g. linear or logistic regression, in a neighbourhood of a given data point to locally approximate the black-box. Its coefficients provide a clear interpretation of the prediction; however, it requires a high fidelity of a surrogate model to be relevant. Even then, we have little evidence that the approximation explains the causal human model. In fact, LIME with a specific loss function, weighting kernel and regularization term becomes the model-agnostic KernelSHAP (Lundberg & Lee, 2017). Break-down is a considerably faster variable attribution explanation focusing on monotonicity at a cost of approximation performance (Y. Liu et al., 2021). These methods provide prediction explanations designed to be understood by machine learning developers, but not necessarily by diverse stakeholders applying them.

As for PVI, it relies on permuting values of a chosen variable and comparing the model's performance loss, which somehow accounts for the importance. It is a model-agnostic method in a sense that it does not assume anything about the machine learning algorithm by only measuring changes in data input with respect to model output. PVI is simple to understand and quite efficient to compute while SHAP value importance bases more on sound theoretic justification instead of heuristics of such kind. Nevertheless in practice, the main disadvantage of all these approaches is their unrealistic assumptions like the independence of variables, or model additiveness in the case of SHAP and LIME. More recently, Aas et al. (2021) extended the KernelSHAP model-agnostic algorithm to handle dependent variables, and provided extensive experiments to support the superiority of their method in a realistic black-box setting. We note that it still relies on estimation utilizing the original data distribution, which in practice is a train or test dataset, leading to fallacies when the distribution drifts, e.g. due to time, stochasticity, or in our case, adversarial reasons.

Various derivations of the above mentioned explanation methods are implemented and available for machine learning practitioners in open-source software, most popularly R packages, e.g. `iml` (Molnar et al., 2018), `DALEX` (Biecek, 2018), `vip` (Greenwell & Boehmke, 2020), and Python packages, e.g. `shap` (Lundberg & Lee, 2017; Lundberg et al., 2020), `aix360` (Arya et al., 2020), `dalex` (Baniecki et al., 2021), `alibi` (Klaise et al., 2021).

## 2.2. Manipulating explainability and fairness

For context, we first discuss the beginning of post-hoc explanations' critique grounded in considerations on the theory-practice mismatch, which was also raised in more general ways (Corbett-Davies & Goel, 2018; Rudin, 2019). Ancona et al. (2018) provided a theoretical unification of the most-popular gradient-based attribution methods for deep neural networks and evaluated them with an introduced *sensitivit-n* property that took perturbation-based explanation as a baseline. Alvarez Melis & Jaakkola (2018) introduced a *self-explaining neural network* with native concept-based interpretability comparing its faithfulness and stability against the explanation maps. It was to give arguments that natively-trained interpretability might be superior to post-hoc explainability. Adebayo et al. (2018) proposed model randomization and data randomization tests to evaluate explanations. The first check investigates the difference in explanations when randomizing a neural network layer of choice, or more generally, randomizing model weights. The second is to compare the explanations' differences when randomizing classification labels (values of the target variable) of data points and retraining the model. Surprisingly, in both cases, little differences in variable attribution explanations were found, which highlighted that correlation does not imply causation. Kindermans et al. (2019) analyzed the *input invariance* property of saliency maps and showed the importance of a reference point (baseline) choice for estimating explanations. Ghorbani et al. (2019) introduced a concept of an adversarial attack on model explanations. These studies raised crucial awareness among machine learning practitioners that explaining models does not entail trust in AI.

Our work relates primarily to recent research done at the junction of the adversarial and explainable machine learning domains. Attacks on model explanations can be categorized with key dimensions like modality of data, type of model and explainability method, the target of an adversary. Overall, the most exploited are variable attribution explanations of deep neural networks; in computer vision oftenly named saliency maps. There are methods for manipulating explanations using gradient-based optimization (Ghorbani et al., 2019; Dombrowski et al., 2019), which resemble a classical concept of adversarial examples. Given a data point as an input, e.g. an image, one adds noise to its values aiming at drastically changing the explanation. The most natural way of perturbing an image classifier is to utilize the gradient of model output with respect to the input. Sinha et al. (2021) use the same approach to manipulate explanations in natural language processing by perturbing words in input text. On the other side, it is possible to fine-tune a neural network to diminish its interpretation possibilities (Heo et al., 2019; Dimanov et al., 2020). This approach requires to change a black-box model to only change its explanations while leaving data and predictions unchanged. Various strategies of fooling are considered, e.g. lowering the attribution of most important variables for each data point, or increasing the attribution of a specific variable (Heo et al., 2019). Anders et al. (2020) extends these results with an in-depth theoretical framework. Moreover, Tomsett et al. (2020) provides a meta-analysis of various evaluation metrics for saliency maps and finds them statistically unreliable and inconsistent. Dimanov et al. (2020) highlight that the same fine-tuning approach of a neural network can be used to hide bias in models by lowering the attribution values for the protected variables in tabular data.

Specifically for tabular data, Slack et al. (2020) introduces a framework for fooling LIME and SHAP of biased black-box models by exploiting the core property of explanations—relying on perturbed data and its distribution. It also assumes that an adversary can substitute a black-box model for providing its unsuspected reasoning, which is the opposite to the data change approach (Ghorbani et al., 2019; Dombrowski et al., 2019). Similarly to variable attribution explanations, it is possible to manipulate the well-established fairness parity metrics (Mehrabi et al., 2021a) via slightly changing the model (Aivodji et al., 2019), selectively choosing the evaluation dataset (Fukuchi et al., 2020; Solans et al., 2020; Mehrabi et al., 2021b), or crafting adversarial examples (Nanda et al., 2021). Lakkaraju & Bastani (2020) conducted a thought-provoking study on misleading effects of manipulated explanations, specifically Model Understanding through Subspace Explanations (MUSE) (Lakkaraju et al., 2019), which provide arguments for why such research becomes crucial to achieve responsibility in machine learning use. Most recently, Slack et al. (2021a) discuss manipulation strategies against counterfactual explanations. Overall, we perceive an apparent research gap—little attack strategies that would be model-agnostic and explanation-agnostic, as most of them rely on various assumptions about either data, models, or explanations. Hereby, this work tackles a problem of providing a unified algorithm for such manipulation. In (Baniecki et al., 2022), we introduced a genetic algorithm to manipulate variable effect explanations by changing the reference dataset, and this work is to extend the method to variable attribution explanations (Baniecki & Biecek, 2022).

## 2.3. Evaluating explanations

As mentioned previously, vulnerabilities and limitations of explainability approaches originates from the work focusing primarily on testing and evaluating explanations. In computer vision, state-of-the-art saliency maps were thoroughly investigated in the context of data and model randomization tests (Adebayo et al., 2018), theoretical sensitivity of explanations (Ancona et al., 2018), and their invariance to input shifts (Kindermans et al., 2019). Hooker et al. (2019) introduced a Remove And Retrain (ROAR) framework, which is a similar approach to model and data randomization tests (Adebayo et al., 2018). It considers substituting image pixels with a baseline, e.g. mean of the pixel value across the dataset, and training several models to assess the specific variables' attribution. The iterative approach finds the most important variables; thus, gives a baseline for evaluating explanations. It is quite a unique approach with many limitations, yet gives us a reasonable approximation and comparison. Adebayo et al. (2020) conduct extensive user-studies to quantify the practical usability of variable attribution explanations by humans. The study categorizes the possible bugs in a machine learning pipeline into data, model, and test-time errors. One of the main results is that current explanations for computer vision have little use in distinguishing mislabeled data points from normal ones. Bhatt et al. (2020) acknowledge the desirable properties of explanations coming from social sciences, which is low sensitivity, high faithfulness, and low complexity. The first corresponds to the robustness of explanations, the second to their accuracy, and the third adheres to humans favouring short and straightforward answers. Apart from vision tasks, explanations for tabular data are evaluated, concluding that an algorithmic aggregation of different methods leads to an improvement across the three fundamental properties.

More recently, research in Human-Computer Interaction focusing on explainability and causability becomes more and more crucial for the domain advancements, which was already highlighted by Miller et al. (2017); Miller (2019). Poursabzi-Sangdeh et al. (2021) aims to measure the interpretability (in this case, causability) capabilities between white-box and black-box of different complexity levels, which effectively became state-of-the-art from the point of research in social sciences. The study considers an apartment price estimation task and compares an interpretable model with black-box in large-scale human studies. Surprisingly, some of the obviously stated hypotheses were not correct under the evidence of the collected data, which again puts an emphasis on evaluating explanations against humans. Jesus et al. (2021) perform a user-study on a real-world fraud detection task with a deployed machine learning model. It concluded with a finding that not necessarily providing more explanations improves human decision accuracy; thus, for successful adoption of black-boxes in financial scenarios, we need to develop explanations suited for non-technical stakeholders. All in all, the landscape of explainable machine learning is vast and varied; Sw. Mishra & Rzeszotarski (2021) evaluates concept-based explanations for through crowd-sourced experiments, Sunder-Samuel et al. (2021) revisit the evaluation of saliency-based explanations on humans, and Neely et al. (2021) revisit the evaluation of variable attribution explanations for natural language processing tasks. Our work relates to the indication of Jia et al. (2021) that there is a correlation between model complexity and explanation quality.

In all this, we see a great challenge of evaluating explanations—lack of ground truth. The problem is conventionally not apparent when evaluating machine learning models for supervised tasks as we have access to the original data labels. However, even if the experts annotate the predictions' explanations for given data points, these will be representative of human perception, not the model behaviour. Zhou et al. (2022) introduce a framework for creating ground truth of explanations in real-world scenarios and apply it in deep learning predictive tasks. The idea is to artificially add features, in this case, watermarks and grammar errors, to images or text that will positively or randomly correlate with the classification labels. Analogous techniques were previously used to evaluate the robustness of deep learning models, e.g. in natural language processing (Rychalska et al., 2019). Explaining accurate models trained to datasets that contain such variables gives insights into the desired properties of explanations, also mentioned by Bhatt et al. (2020). Y. Liu et al. (2021) provide the first ground-truth benchmark for variable attributions in tabular data. The constructed theoretical framework and software allows to sample experimental datasets from the known distributions; this knowledge can be then utilized to accurately compute the conditional distributions of variables and various expected values used in explanation estimation.

## 2.4. Defense against explanation manipulation

Further related, sparse, and the least diverse is work concerning the development of robust explanations (Sa. Mishra et al., 2021), which would exhibit input stability, resistance to the adversarial attacks, and high causability capabilities. Various enhancements to training deep neural networks are proposed in spirit of improving the robustness of the aforementioned saliency maps (Alvarez Melis & Jaakkola, 2018; Wang et al., 2020; Boopathy et al., 2020). The most straightforward approach is to aggregate various explanation maps, which proves to be robust against adversarial attacks that are proven to manipulate a single one (Rieger & Hansen, 2020). Dombrowski et al. (2022) revisit the manipulation strategies of Dombrowski et al. (2019), and provide algorithms that indeed improve the robustness of explanations for deep neural networks.

To this date, there is little work on enhancing the robustness of state-of-the-art explanations for machine learning models trained on tabular data (Sa. Mishra et al., 2021). Lakkaraju et al. (2020) proposes a framework that aims to construct explanations with high fidelity against the worst-case scenario over a set of adversarial perturbations. Both Zhao et al. (2021) and Slack et al. (2021b) propose to incorporate a bayesian approach into the simple surrogate model in LIME, which allows to incorporate an expert knowledge into the construction of explanations, as well as provides a native measure of the explanation's uncertainty. The latter can be used to detect the potential adversary. There is research to be done on the robustness of other widely-used explanations like PVI or Accumulated Local Effects (Apley & Zhu, 2020).

## 2.5. Summary

To better survey the current state of the art in adversarial and explainable machine learning, we propose a taxonomy incorporating the following representative dimensions of methodologies:

1. **Domain**: explainability, fairness.
2. **Focus**:
   - attack, often called manipulation or fooling,
   - defense, *including* proposing robust explanations,
   - evaluation of explanations related to their vulnerabilities.
3. **Change**:
   - model, e.g. in case of adversarial fine tuning,
   - data, e.g. in case of adversarial examples,
   - explanation, e.g. in case of introducing robust explanations,
   - both, e.g. in case of adversarial training.
4. **Data modality**: image, tabular, text.
5. **Model type**:
   - model-agnostic, meaning working with black-box models,
   - model-specific methods, e.g. suited for neural networks through using gradients.
6. **Explanation type**:
   - local, explaining the prediction,
   - global, explaining the overall model's behaviour, which includes fairness measures.

The definition of local and global explanations follows the description of Molnar (2020); Biecek & Burzykowski (2021), distinct from the local and global attributions considered by Ancona et al. (2018); Yeh et al. (2019). Please note that we report elements like data modality, model and explanation type as considered by the referenced publications, while some methods may be applicable to other data or models as well. We acknowledge the fact that such dimensions might be a limited simplification, but this is only a gentle attempt to systemize the current state of knowledge.

Table 2.1 summarises our findings based on 43 articles, from which we have two observations: (1) most works are related to local explanations of neural network models (32, 74%) used for image classification (27, 63%), and (2) there is an evident research gap: little number of works considering model-agnostic methods of attacking local explanations for tabular data (2, 6%). Notably, we exclude from the table the user study of Poursabzi-Sangdeh et al. (2021), which considers evaluating white-box interpretability vs black-box models (Rudin, 2019), and not an adversarial manipulation, or the explanations' vulnerabilities, in general. To the best of our knowledge, the closest work to ours changes the black-box model in an adversarial manner to fool SHAP (Slack et al., 2020). In contrast, we explore the possibility of altering the reference values (baseline dataset) used for the estimation of explanations.

Table 2.1: Categorized list of related articles considering various vulnerabilities of explanations.

| Reference | Domain | Focus | Change | Data modality | Model type | Explanation type |
|---|---|---|---|---|---|---|
| (Ancona et al., 2018) | explainability | evaluation | data | image, text | neural network | local |
| (Alvarez Melis & Jaakkola, 2018) | explainability | defense | model | image, tabular | neural network | local |
| (Adebayo et al., 2018) | explainability | evaluation | model, data | image | neural network | local |
| (Ghorbani et al., 2019) | explainability | attack | data | image | neural network | local |
| (Aivodji et al., 2019) | explainability, fairness | attack | data | tabular | black-box | global |
| (Kindermans et al., 2019) | explainability | attack | data | image | neural network | local |
| (Woods et al., 2019) | explainability | defense | model, data | image | neural network | local |
| (Heo et al., 2019) | explainability | attack | model | image | neural network | local, global |
| (Dombrowski et al., 2019) | explainability | attack | data | image | neural network | local |
| (Yeh et al., 2019) | explainability | defense | explanation | image | neural network | local |
| (Chen et al., 2019) | explainability | defense | model | image | neural network | local |
| (Hooker et al., 2019) | explainability | evaluation | model, data | image | neural network | local |
| (Dimanov et al., 2020) | explainability, fairness | attack | model | tabular | neural network | local, global |
| (Slack et al., 2020) | explainability, fairness | attack | model | tabular | black-box | local |
| (Lakkaraju et al., 2019) | explainability, fairness | attack | model | tabular | black-box | global |
| (Fukuchi et al., 2020) | fairness | attack | data | tabular | black-box | global |
| (Tomsett et al., 2020) | explainability | evaluation | — | image | neural network | local, global |
| (Anders et al., 2020) | explainability | attack | data | image, tabular | neural network | local, global |
| (Rieger & Hansen, 2020) | explainability | defense | data | image | neural network | local |
| (Boopathy et al., 2020) | explainability | defense | model, data | image | neural network | local |
| (Lakkaraju et al., 2020) | explainability | defense | explanation | tabular | black-box | local |
| (Kuppa & Le-Khac, 2020) | explainability | attack | data | tabular | neural network | local |
| (Zhang et al., 2020) | explainability | attack | data | image | neural network | local |
| (Solans et al., 2020) | fairness | attack | model, data | tabular | neural network | global |
| (Wang et al., 2020) | explainability | defense | model | image | neural network | local |

Table 2.1: (Continued) Categorized list of related articles considering various vulnerabilities of explanations.

| Reference | Domain | Focus | Change | Data modality | Model type | Explanation type |
|---|---|---|---|---|---|---|
| (Adebayo et al., 2020) | explainability | evaluation | model, data | image | neural network | local |
| (Bhatt et al., 2020) | explainability | explanation | — | image, tabular | neural-network | local |
| (Mehrabi et al., 2021b) | fairness | attack | model, data | tabular | black-box | global |
| (Nanda et al., 2021) | fairness | evaluation | model, data | image | neural network | global |
| (Hanawa et al., 2021) | explainability | evaluation | model, data | image, text | neural network | local |
| (Zhao et al., 2021) | explainability | defense | explanation | image, tabular | black-box | local |
| (La Malfa et al., 2021) | explainability | defense | explanation | text | neural network | local |
| (Lin et al., 2021) | explainability | evaluation | model, data | image | neural network | local |
| (Jia et al., 2021) | explainability | evaluation | model | image | neural network | local |
| (Sinha et al., 2021) | explainability | attack | data | text | neural network | local |
| (Aïvodji et al., 2021) | explainability, fairness | attack | model | tabular | black-box | global |
| (Slack et al., 2021a) | explainability | attack | data | tabular | neural network | local |
| (Slack et al., 2021b) | explainability | defense | explanation | image, tabular | black-box | local |
| (Dombrowski et al., 2022) | explainability | defense | model | image | neural network | local |
| (Tang et al., 2022) | explainability | defense | model | image | neural network | local |
| (Zhou et al., 2022) | explainability | evaluation | model, data | image, text | neural network | local |
| (Baniecki et al., 2022) | explainability | attack | data | tabular | black-box | global |
| (Baniecki & Biecek, 2022) | explainability | attack | data | tabular | black-box | local, global |

# 3. Methods

In this Chapter, we first describe the data, models, and explanations involved in our study. Then, Section 3.5 introduces the manipulation algorithm and Section 3.4 defines evaluation measures.

## 3.1. Data

The choice of datasets is crucial for appropriate evaluation of machine learning methodology. We can categorize types of data sources most commonly used for such experiments into four groups:

1. synthetic generated data, e.g. the XAI-Bench library (Y. Liu et al., 2021),
2. open data repositories, e.g. the UCI machine learning repository (Dua & Graff, 2017),
3. benchmarks, e.g. the OpenML-CC18 benchmark (Bischl et al., 2021),
4. real-world applications, e.g. the MIMIC-4 database (Johnson et al., 2021).

Synthetic datasets are created on demand and most easily validate methods in custom settings. Their use is beneficial for testing methods and general benchmarking; however, they do not present applicability of the methods in real-world settings. This is why machine learning practitioners collaborate on sharing data in open repositories. Yet again, recent critique is that such datasets are too simple, rapidly become obsolete, and do not exactly resemble real-world problems (Paullada et al., 2021). For practical reasons, multiple datasets are often joined together to define particular benchmarks used for more standardized comparisons between machine learning methods. To this day, the closest to real-world applications are datasets originating from research and development initiatives, and works specifically focused on gathering, cleaning, describing, and sharing datasets. Such critique motivated the creation of the *NeurIPS 2021 Datasets and Benchmarks Track.*[1]

**Synthetic data.** Taking the above discourse into consideration, we rely on both synthetic generated data and a real-world application in our experiments. The first allows to rapidly validate hypotheses of interest by setting custom number of observations, variables, the ratio of noise in data, and relationships between the variables and target itself. The second presents the general applicability of the methods in use cases of interest. Specifically, we use the eXplainable AI Benchchmark (XAI-Bench) library (Y. Liu et al., 2021) for benchmarking explanations, denoted by `xaibench`, to create the following 2 tasks:

[1] More information is available at `https://neurips.cc/Conferences/2021/CallForDatasetsBenchmarks`.

- `gaussian`: Multivariate Gaussian variables with a nonlinear additive binary classification target, where variables $X$ are generated with parameters $\mu = 0,\ \sigma = 1,\ \rho = 0.5$. Target variable $y$ is first defined by the formula

$$y \coloneqq -\sin(2X_1) + |X_2| + X_3 + \exp(-X_4) + \cos(3X_5), \tag{3.1}$$

  and then standardized by subtracting mean and dividing by standard deviation. It is contaminated with Gaussian noise $N(0, \frac{1}{4})$, and finally transformed into a binary target with $y \coloneqq \mathbb{1}(\frac{\exp(y)}{1+\exp(y)} > 0.5)$.

- `mixture`: Mixture of multivariate Gaussian variables with a nonlinear additive regression target, where variables $X$ are generated with $\mu = [-3, 3],\ \sigma = [1, 1],\ \rho = 0.5$. Target variable $y$ is defined by a similar formula

$$y \coloneqq -\sin(2X_1) + |X_2| + X_3 + \exp(-X_4) + \cos(3X_5) + \epsilon, \tag{3.2}$$

  where $\epsilon \sim N(0, \frac{1}{4})$, and then standardized.

These datasets were previously used in explainable machine learning research to analyze the additiveness and linearity of explanations, see for instance (Chen et al., 2018). For each task, we create 3 subtasks distinguishing the complexity of the problem:

- `small`: $X$ contains 10 variables, in which $X_1$–$X_5$ are informative by contributing to the target $y$ while $X_6$–$X_{10}$ remain as noise,
- `medium`: $X$ contains 25 variables, in which 10 are informative – we transform $X_6$–$X_{10}$ in the same way as $X_1$–$X_5$, and $X_{11}$–$X_{25}$ remain as noise,
- `large`: $X$ contains 100 variables, in which 25 are informative.

This concludes with 6 synthetic datasets in total; each has 1000 observations in the training set and 200 observations in the validation set, which is used for estimating model performance and as a baseline for computing explanations.

**Medical data.** Explainable machine learning methods become extensively applied in biomedicine (Holzinger et al., 2019; Lundberg et al., 2020; Markus et al., 2021); thus, for a representative example of real-world applications, we use the MIMIC-4 database (Johnson et al., 2021). Medical Information Mart for Intensive Care (MIMIC) is a large, freely-available database comprising deidentified health-related data from patients who were admitted to the critical care units of the Beth Israel Deaconess Medical Center. These are medical records of over 60 thousand patients admitted between 2002-2019. MIMIC-4 supersedes the MIMIC-3 database (Johnson et al., 2016), which is highly referenced by machine learning literature, e.g. Bhatt et al. (2020) use the data when evaluating explanations. MIMIC-4 is not publicly available due to containing potentially sensitive information, rather free access to these protected datasets is granted upon a reasonable request. Specifically, we utilize it's preprocessed subset defined as the metaMIMIC dataset (Woźnica et al., 2022), which is tabular data with 172 explanatory variables and 12 target variables for 34 925 patients.[2]

[2] Code used to create the dataset is openly available at `https://github.com/ModelOriented/metaMIMIC`.

The goal is to predict the occurrence of a specific disease based on the measurements made during the patient's hospital stay. Table 3.1 describes the explanatory variables, which are divided into two groups: (1) age, gender weight, and height of the patient, and (2) 56 medical measurements where each of them generates 3 final explanatory variables—these are minimum, average and maximum values over the time series of the patient's hospital stay. In general, the data contains some missing values and outliers, which makes it a challenging task for machine learning algorithms. We further elaborate on this fact in Section 3.2. The metaMIMIC dataset contains 12 targets for binary classification, from which we arbitrarily choose 3 targets between 25% and 40% prevalence to cover similar medical scenarios for simplicity. For a deeper description of the database and tasks, we refer the reader to the work of Woźnica et al. (2022).

We denote the metaMIMIC dataset by `mimic4` and choose the following diseases corresponding to the International Classification of Diseases (ICD)[3] to create the final tasks:

- `anemia`: Anemia disease represented as the ICD-9 code number 280-285 and ICD-10 code number D60-D64, which has a 35.9% prevalence among patients,
- `diabetes`: Diabetes disease represented as ICD-9:249-250 and ICD-10:E08-E13, which has a 25.3% prevalence among patients,
- `heart`: Ischematic heart disease represented as ICD-9:410-414 and ICD-10:I20-I25, which has a 32.8% prevalence among patients.

For each task, we create 2 subtasks distinguishing the complexity of the problem:

- `small`: the dataset contains the top 10 most important explanatory variables out of 172, where this filtering step is made based on a variable importance measurement coming from a baseline XGBoost model (Chen & Guestrin, 2016),
- `large`: the dataset contains the top 100 most important explanatory variables.

We choose to omit the `medium` subtask in this case, which allows us to study more disease predictive tasks. This concludes with 6 real-world datasets in total; each contains about 17 462 observations in the training set, 17 462 observations in the test set, which is used for estimating model performance, and 1000 observations in the validation set, which is used as a baseline for computing explanations (and in our case is a subset of the test set).

Overall, for a balanced overview between synthetic and real-world machine learning problems, we propose to evaluate explanations on 12 distinct predictive tasks, which vary in the number of variables, observations, and overall predictive complexity. Please note that, while most of the tasks consider a binary classification, we believe the results should generalize to other predictive tasks, e.g. regression. We include one of such datasets in the benchmark to show the general applicability of our methodology. On the size of the validation set, in the case of synthetic data we know that train and test sets are generated from the same distribution; hence, there is no need for a larger validation of model performance. In case of `mimic4`, we split the data in half and use these parts as train and test sets; yet, we need to restrict the validation set's size for a reasonable computational cost of explanations estimation. We further discuss the quantity of baseline observations in Section 3.3.

[3] See ICD by WHO at https://www.who.int/standards/classifications/classification-of-diseases.

Table 3.1: Description of explanatory variables in the metaMIMIC dataset, which includes data from three MIMIC-4 database tables: patients, chartevents, and labevents. Aggregation based on statistics denotes using minimum, average and maximum values over the time series, which become the final explanatory variables.

| ID | Variable | Category | Table | Aggregation | Missing |
|---:|---|---|---|---|---:|
| — | Age | General | patients | — | 0% |
| — | Gender | General | patients | — | 0% |
| 226512 | Admission Weight (Kg) | General | chartevents | first value | 0.1% |
| 226730 | Height (cm) | General | chartevents | first value | 45.3% |
| 226253 | SpO2 Desat Limit | Alarms | chartevents | statistics | 0.7% |
| 220228 | Hemoglobin | Labs | chartevents | statistics | 2.3% |
| 220546 | WBC | Labs | chartevents | statistics | 2.3% |
| 225624 | BUN | Labs | chartevents | statistics | 2.1% |
| 227073 | Anion gap | Labs | chartevents | statistics | 2.1% |
| 227457 | Platelet Count | Labs | chartevents | statistics | 2.3% |
| 227465 | Prothrombin time | Labs | chartevents | statistics | 12.5% |
| 227466 | PTT | Labs | chartevents | statistics | 13.0% |
| 220739 | GCS - Eye Opening | Neurological | chartevents | statistics | 0.2% |
| 223900 | GCS - Verbal Response | Neurological | chartevents | statistics | 0.2% |
| 223901 | GCS - Motor Response | Neurological | chartevents | statistics | 0.2% |
| 223791 | Pain Level | Pain/Sedation | chartevents | statistics | 10.7% |
| 220210 | Respiratory Rate | Respiratory | chartevents | statistics | 0.1% |
| 220277 | O2 saturation pulseoxymetry | Respiratory | chartevents | statistics | 0.1% |
| 223834 | O2 Flow | Respiratory | chartevents | statistics | 24.3% |
| 220045 | Heart Rate | Routine Vital Signs | chartevents | statistics | 0.0% |
| 220179 | Non Invasive Blood Pressure systolic | Routine Vital Signs | chartevents | statistics | 1.1% |
| 220180 | Non Invasive Blood Pressure diastolic | Routine Vital Signs | chartevents | statistics | 1.1% |
| 223761 | Temperature Fahrenheit | Routine Vital Signs | chartevents | statistics | 1.6% |
| 224054 | Braden Sensory Perception | Skin - Assessment | chartevents | statistics | 0.6% |
| 224055 | Braden Moisture | Skin - Assessment | chartevents | statistics | 0.6% |
| 224056 | Braden Activity | Skin - Assessment | chartevents | statistics | 0.6% |
| 224057 | Braden Mobility | Skin - Assessment | chartevents | statistics | 0.6% |
| 224058 | Braden Nutrition | Skin - Assessment | chartevents | statistics | 0.6% |
| 224059 | Braden Friction/Shear | Skin - Assessment | chartevents | statistics | 0.6% |
| 50802 | Base Excess | Blood Gas | labevents | statistics | 34.6% |
| 50804 | Calculated Total CO2 | Blood Gas | labevents | statistics | 34.6% |
| 50813 | Lactate | Blood Gas | labevents | statistics | 33.2% |
| 50818 | pCO2 | Blood Gas | labevents | statistics | 34.6% |
| 50820 | pH | Blood Gas | labevents | statistics | 32.5% |
| 50821 | pO2 | Blood Gas | labevents | statistics | 34.6% |
| 50861 | Alanine Aminotransferase (ALT) | Chemistry | labevents | statistics | 38.0% |
| 50863 | Alkaline Phosphatase | Chemistry | labevents | statistics | 38.7% |
| 50868 | Anion Gap | Chemistry | labevents | statistics | 0.5% |
| 50878 | Asparate Aminotransferase (AST) | Chemistry | labevents | statistics | 37.9% |
| 50882 | Bicarbonate | Chemistry | labevents | statistics | 0.5% |
| 50885 | Bilirubin, Total | Chemistry | labevents | statistics | 38.7% |
| 50893 | Calcium Total | Chemistry | labevents | statistics | 3.2% |
| 50902 | Chloride | Chemistry | labevents | statistics | 0.5% |
| 50912 | Creatinine | Chemistry | labevents | statistics | 0.4% |
| 50931 | Glucose | Chemistry | labevents | statistics | 0.5% |
| 50960 | Magnesium | Chemistry | labevents | statistics | 0.9% |
| 50970 | Phosphate | Chemistry | labevents | statistics | 3.1% |
| 50971 | Potassium | Chemistry | labevents | statistics | 0.5% |
| 50983 | Sodium | Chemistry | labevents | statistics | 0.5% |
| 51006 | Urea Nitrogen | Chemistry | labevents | statistics | 0.4% |
| 51221 | Hematocrit | Hematology | labevents | statistics | 0.5% |
| 51222 | Hemoglobin | Hematology | labevents | statistics | 0.5% |
| 51248 | MCH | Hematology | labevents | statistics | 0.5% |
| 51249 | MCHC | Hematology | labevents | statistics | 0.5% |
| 51250 | MCV | Hematology | labevents | statistics | 0.5% |
| 51265 | Platelet Count | Hematology | labevents | statistics | 0.5% |
| 51277 | RDW | Hematology | labevents | statistics | 0.5% |
| 51279 | Red Blood Cells | Hematology | labevents | statistics | 0.5% |
| 51301 | White Blood Cells | Hematology | labevents | statistics | 0.5% |
| 51491 | pH | Hematology | labevents | statistics | 41.5% |

## 3.2. Models

Data quality lies at the core of effective data analysis, and so is the choice of machine learning model algorithms.Nowadays, there is a discourse among machine learning practitioners about the improving effectiveness of neural networks on tabular data (Arik & Pfister, 2021; Gorishniy et al., 2021); however, the tree-ensemble models are already proven to work best on average in classical machine learning settings (Shwartz-Ziv & Armon, 2022). Gradient Boosting Decision Trees (GBDT) (Chen & Guestrin, 2016; Ke et al., 2017) sequentially creates decision trees of varied structure and ensembles their prediction at the end to achieve best possible predictive performance. In contrast, Feedforward Neural Networks (FNN) iteratively learn weights in a specific model architecture with gradient descent (see a textbook by Goodfellow et al. (2016) and the references given there). Both of these model families are considered black-box'es, which need to be explained, for example with SHAP. We discuss the choice of models with respect to the considered explanation approaches in Section 3.3. Taking the wide adoption of GBDT and FNN in modern machine learning, we use the following models in our study:

- `GBDT`: Gradient boosting decision trees trained using the `xgboost` Python package (Chen & Guestrin, 2016) with the fixed parameters: `learning_rate` = 0.1, (row) `subsample` = 0.7, `colsample_bytree` = 0.7, `colsample_bynode` = 0.7, and two changing parameters accounting for model complexity: `max_depth` = $3, 4, 6, 9$ and `n_trees` = $100, 200, 400, 700$. We denote a GBDT model with 200 trees of depth 4 by `GBDT:4:200`.

- `FNN`: Feedforward neural networks trained using the combination of `poutyne` and `torch` Python packages (Paszke et al., 2019; Paradis et al., 2020) with two changing parameters accounting for model complexity: `layer_count` = $1, 2, 3, 4$ and `neuron_count` = $16, 32, 64, 128$, and the fixed parameters: ReLU activation function (Nair & Hinton, 2010), Batch normalization after each activation layer (Ioffe & Szegedy, 2015), Adam optimizer (Kingma & Ba, 2015) with default parameters `learning_rate` = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We denote a FNN model consisting of 3 layers with 64 neurons each by `FNN:3:64`.

All models use cross entropy loss for classification or mean squared error for regression tasks. We describe the used model performance measures in Section 3.4. Overall, for a broad overview of machine learning algorithms, we propose to evaluate explanations on 32 distinct predictive models, which vary in complexity and predictive performance.

**On missing values in the `mimic4` data.** We acknowledge that decision trees are designed to be able to train with missing values, and this property is propagated to GBDT models. In the case of FNN, missing values imputation methods of varied complexity are available to work around this problem (Woznica & Biecek, 2020), e.g. imputation with average, median, or by predicting the value from the data itself. In the case of this study specifically, we decide to impute missing values with 0 for through the following thought process: (1) from the perspective of data, it is probable that inputing these values with average would change the underlying information, e.g. a patient did not have a medical examination but has the result, (2) in training FNN, for input value $x$ and initial weight value $w$, the operation $x \cdot w$ always returns 0, leading to effectively removing missing variable $x$ from the equation, (3) in explaining FNN with SHAP, it is not obvious that values other than 0 would be advantageous from the interpretation perspective, and (4) our manipulation algorithm and evaluation measures are invariant to missing values encoded as 0, which is a useful property that we further discuss in Section 3.5. Overall, this choice of missing value imputation strategy is important to notice as a possible limitation, but it should not affect on the final results. We restrain from imputing missing values when interacting with GBDT models.

## 3.3. Explanations

Shapley value framework can in theory be applied to any machine learning predictive problem. In practice, several assumptions can be made when computing SHAP values like variable independence and model linearity (see equations 9-12, Lundberg & Lee, 2017), meaning that the explanations are modeled through the additive composition of variable attributions. The model-agnostic approximation of SHAP values is named KernelSHAP method, which stands from providing a specific kernel function to the linear surrogate model fitted in the LIME method (Ribeiro et al., 2016). Overall, both LIME and KernelSHAP involve randomness in estimating explanation through bootstrapping and perturbing data used as a *baseline* distribution (often called *reference value*). In our case, since we specifically consider GBDT and FNN models, it is beneficial to consider the model-specific estimators of SHAP values, which we now elaborate on.

GradientSHAP combines the Integrated Gradients explanation (Sundararajan et al., 2017) for variable attributions of neural networks with the SmoothGrad method (Smilkov et al., 2017) under the assumptions mentioned above to effectively approximate SHAP values for neural networks; this approach was introduced by Yeh et al. (2019). Intuitively, Integrated Gradients measures an accumulation of neural network's output gradients computed at a grid of points between the input and baseline (hence the name of the method comes from an integral). SmoothGrad improves the robustness of such attribution by aggregating with mean the explanations computed for multiple perturbed variations of the input. Furthermore, two types of such attributions are considered in the literature: local and global, *not to be mistaken with the terms local and global explanations*. As described by Ancona et al. (2018), an illustrative example of local attribution would be simply a coefficient in the linear regression model, while an alternative global attribution would be the same coefficient multiplied by an input value. This might be unintuitive from the perspective of local and global explanations where one could state the opposite. Yet, local attributions aim to measure how the output of the network changes *locally*, e.g. for small perturbations of the input, while global attributions describe the actual effect of a variable and should sum to the model's prediction.

We use a GradientSHAP implementation from the `captum` Python package (Kokhlikyan et al., 2020), which is the most popular tool for explaining `torch` neural networks. For context to our methodology, it is worth discussing the important parameters available in the used (current) version `0.5.0` of the method's API:[4]

1. `inputs` – Observations for which one computes SHAP attribution values.
   **Comment:** It can be any number of inputs ranging from a single observation, for which one wants to compute local explanation of a prediction, to a full validation set, where one can approximate global variable importance, e.g. by computing a mean of the absolute attribu-

[4] The full API reference for GradientSHAP is available at `https://captum.ai/api/gradient_shap`.

tion values across the data. Anyhow, the main goal of this study is to manipulate and evaluate local and global SHAP, so for each task × model pair, e.g. `anemia:large:FNN:3:64`, we choose an inbetween explanation of $K$ observations as a potential target. The process of choosing these $K$ observations is consistent across all the experiments: we first compute a matrix of attribution values for the whole validation set, and then use a naive neighbour search algorithm to find $K-1$ nearest neighbours for each observation (in the attribution space). We employ the euclidean distance, which is applied without data standardization as the attributions share a similar domain by design (they sum to the prediction). Finally, we choose a central observation with its $K-1$ neighbours where the sum of distances is the lowest. For convenience, we set $K=10$ across all the experiments in this work while manipulating this quantity is mentioned as a potential future work in Chapter 5. In conclusion, we target semi-global explanations for a clique of $K$ inputs based on their attribution values, which omits the following obstacles: (1) arbitrarily choosing only a single local explanation, (2) high computational cost when considering all observations, (3) relying on a possibly wrong choice of the standardization method for performing such a filtering based on data values.

2. `baselines` – Data used as reference values for computing expected values.
**Comment:** Choice of a proper baseline for estimating explanations is a vast topic that has itself initiated a discourse in explainable machine learning research, which we cover in Chapter 2. In the case of this study, we assume existence of a validation set, possibly coming from the test set, which is used as a baseline in all the explanation scenarios. This is a subset ranging between 200-1000 observations, which is a widely used standard in explanation computation and often depends on the size of training set and model complexity. We believe the size of 1000 is an improvement over about 100 observations used in our previous work (Baniecki & Biecek, 2022), as naturally, increasing the baseline size affects the computational cost.

3. `n_samples` – Number of randomly generated samples per observation in `inputs`.
**Comment:** The default value is set to 5. We increase it to 11 across all the experiments with the aim to lower the randomness and increase the stability of explanation estimation.

4. `stdevs` – Standard deviation of the Gaussian noise used to generate the random samples.
**Comment:** Note that in its current implementation, this parameter can only include one value per observation, which effectively leads to adding noise from the same distribution to each variable. This simplification might come from the fact that, conventionally, GradientSHAP is used to explain deep neural networks for computer vision or natural language processing tasks, where inputs very often are standardized in some way, e.g. pixels, word embeddings; thus, adding Gaussian noise with similar magnitude for each variable makes sense. Since we focus on tabular data with varied variables' distributions instead, we *modify* the original implementation of `captum` to allow passing a vector of standard deviations per observation. In fact, it becomes one vector of standard deviations for all the considered observations, as these come from the same distribution. In what follows, the default value of `stdevs` is set to 0, while we set it to a value proportional to the standard deviation of the variable estimated from the validation set; we multiply each standard deviation by an arbitrary ratio factor of $\frac{1}{9}$, which relates to the mutation ratio described in Section 3.5.

5. target – Output (column) index for which gradients are computed.
   **Comment:** Naturally, we set this parameter to 0 in the case of regression and 1 in the case of binary classification (for the class of interest).

6. multiply_by_inputs – Indicator switching between computing local or global attributions.
   **Comment:** We aim to manipulate and evaluate global attributions as they resemble conventional SHAP values, which have a desired property of summing up to the model's prediction; hence, we set this parameter to True.

This concludes the theory and practice behind the explanations for FNN. The second model-specific estimator of SHAP values is TreeSHAP, a fast and exact approximation method for tree models and ensembles of trees introduced by Lundberg et al. (2020). At its core, the algorithm utilizes the rule-like representation of trees, and its property of efficiently storing information about the data distribution with the predicted outcomes in the trees' leaf nodes. To make an inference, usual tree-ensemble frameworks store paths of variable splits from the root node to leaves, fractions of training observations going down each path, as well as the expected prediction. Moreover, tree-ensembles are based on aggregating weak learners' predictions, so it is possible to compute SHAP values for each tree and then average the attributions over all the trees (with weights if needed). Compared to exact Shapley value estimation, TreeSHAP reduces the computational complexity from $O(2^{|V|}TL)$ to $O(D^2TL)$, where $|V|$ is the number of variables, $T$ is the number of trees, $L$ is the maximum number of leaves in any tree, and $D$ the maximal depth of any tree. Note that the factor $2^{|S|}$ comes from the Equation 2.1 where one needs to take into account all the possible subsets of variables. How is it possible that the TreeSHAP computes in polynomial time instead of exponential? The key factor is to keep track of the fractions of variable subsets going down into each of the leaves at the cost of $O(D^2 + |V|)$ memory complexity, which is possible in the case of decision trees. We specifically focus on the independent TreeSHAP algorithm, which uses background reference samples to estimate expectations and runs in $O(RTL)$ time, where $R$ is the number of samples. For a deeper insight into the TreeSHAP algorithm's variations, we refer the reader to the original paper introducing the method (Lundberg et al., 2020).

We use the original implementation of TreeSHAP, which is available in the very popular shap Python package (Lundberg & Lee, 2017) and works well with xgboost models. Since we target a practical estimator of explanation, it is worth mentioning a few parameters available in the used (current) version 0.40.0 of the method's API:[5]

1. data – Data used as reference values for computing expected values.
   **Comment:** Similarly to GradientSHAP, one can use a validation set as a background reference samples. Crucially *can* but does *not* have to. As mentioned before, we can automatically use the training samples as our background dataset, whose information is practically encoded in the decision tree structure through split and leaf nodes. This is a faster and default setting, which we intentionally break in our study by always supplying a validation set as reference values. More often than not, one should evaluate models in varied test settings rather than relying on the distribution used to construct the model itself. Citing the shap documentation *"Anywhere from 100 to 1000 random background*

[5] The full API reference for TreeSHAP is available at https://shap.readthedocs.io/en/latest/generated/shap.explainers.Tree.

*samples are good sizes to use."* so is another argument toward the 200-1000 size of our validation sets. All in all, our work is partially to raise awareness about the importance of baseline choice for estimating explanations.

2. `feature_perturbation` – Indicator switching between relying on training distribution, or background reference samples for computing conditional expectations.
   **Comment:** This parameter seems reduntant as it works interchangeably with `data`, and we set it to the default value of `interventional`.

3. `model_output` – Type of model output to be explained by the method.
   **Comment:** We set this parameter to `raw` in the case of regression and `probability` in the case of binary classification, as it is more informative than log odds ratio.

## 3.4. Evaluation measures

There are well-established rules of evaluating machine learning models; however, evaluating explanations poses challenges covered by us in Chapter 2. We use the following conventional measures to validate model performance:

- In binary classification tasks, we rely on the $F_1$ score and Area Under the Receiver Operating Characteristic Curve (AUC ROC, which we shorten to $AUC$ when the context is clear). Although the accuracy measured as a fraction of accurate predictions for a given probability threshold is the most popular measure of classification performance, we omit its use. This is because the considered predictive tasks are imbalanced in a sense of class frequency, which favours the use of $F_1$ score defined as

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)}, \tag{3.3}$$

  where $TP$ denotes true positives, $FP$ denotes false positives, and $FN$ denotes false negatives. $F_1$ omits measuring true negatives $TN$, which become the largest group in inbalanced classification. For a broader context, we report $AUC$ measured by calculating an integral under the ROC curve, which shows the true positive rate $\frac{TP}{TP+FN}$ against the false positive rate $\frac{FP}{FP+TN}$ at different classification thresholds between 0 and 1.

- In regression tasks, we rely on the $R^2$ measure defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}, \tag{3.4}$$

  where $y$ denotes true target values, $\overline{y}$ is their mean, and $\hat{y}$ denotes predicted values. It should range between 0 and 1 (negative values are possible when the model is worse than a baseline that always predicts $\overline{y}$); thus, can be compared between models fitted to different predictive tasks. Additionally, we compute Mean Squared Error (MSE) defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{3.5}$$

**Measuring explanations.** In the next Section 3.5, we introduce an algorithm to force a change in explanations. There are various ways of measuring this phenomenon. A first and natural way is to calculate the distance between an original and manipulated explanation like presented in Figure 1.1. Let us denote by $a$ and $b$ the vectors of original and manipulated explanations respectively with $a_j$ being the absolute SHAP value of variable $j$ averaged for $K$ observations (see the definition of `inputs` in Section 3.3):

$$a_j = \frac{1}{K}\sum_{i=1}^{K} |\phi_j(f,\ x^i)|. \tag{3.6}$$

Hence $\forall_j\ a_j \geq 0 \wedge b_j \geq 0$. We can consider the following distance measures:

$$d_1(a,b) = \sum_{j=1}^{|V|} |a_j - b_j|, \tag{3.7}$$

$$d_2(a,b) = \sum_{j=1}^{|V|} \left(a_j - b_j\right)^2. \tag{3.8}$$

The only caveat is how to normalize these distances to be able to compare them between various tasks. We propose the following two definitions of manipulation effectiveness based on $d_1$:

$$\overline{ME}(a,b) = \frac{1}{|V|}\sum_{j=1}^{|V|} \frac{|a_j - b_j|}{a_j + \epsilon}, \tag{3.9}$$

$$\underline{ME}(a,b) = \frac{\sum_{j=1}^{|V|} |a_j - b_j|}{\epsilon + \sum_{j=1}^{|V|} a_j}, \tag{3.10}$$

where $\epsilon$ is a small number added to prevent division by zero. We arbitrarily chose $d_1$ over $d_2$ as it has a more natural interpretation, and we don't need this function to be differentiable, which would normally speak in favour of $d_2$. In our practice, both manipulation effectiveness measures are alike; however, minimal changes to SHAP values of unimportant variables where $\exists_j\ a_j \simeq 0 \wedge b_j \gg 0$ has a potential to increase $\overline{ME}$ too much, which makes $\underline{ME}$ more robust. Therefore, from now on, we use $\underline{ME}$ to measure manipulation effectiveness and denote it as $ME$ where the context is clear.

For example in Figure 1.1, we report the following measure values: $d_1 = 3.55$, $d_2 = 2.73$, $\overline{ME} = 8.36$ for $\epsilon = 0.01$, and $\underline{ME} = 0.82$ for $\epsilon = 0.001$. From this, there are two observations: First, we delivered an anecdotal proof that shows an issue with $\overline{ME}$ – although a large change in the variable age may be detrimental, we would prefer manipulation effectiveness to lay between 0 and 1 in most scenarios. Second, even relatively large values of $d_1$ and $d_2$ may not be impactful from the perspective of wrongly explaining the model's reasoning by the end-user. From the side of various stakeholders applying explanations in practice, the ranking of variables in an explanation is often more important than the raw values of attributions. Therefore, we propose to measure a change in the ranking of SHAP values between the original and manipulated explanation. We refer the reader to Kruskal (1958) for a broader review of ordinal measures of association. Let us first define a Kendall rank correlation coefficient (Kendall, 1938), commonly referred to as Kendall's $\tau$, which was used before for measuring explanations (Chen et al., 2019).

**Definition 3.1 (Kendall rank correlation coefficient (Kendall, 1938)).** Let $(X_i, Y_i)$, $i = 1 \dots n$, be independent and identically distributed (iid) random vectors. Further, let $(i, R_i)$, $i = 1 \dots n$, be paired rankings, where $R_i$ is the rank of $Y$ whose corresponding $X$ has rank $i$ (ties are neglected for simplicity). A Kendall's tau is defined as

$$\tau(X,Y) = \frac{2}{n(n-1)} \sum_{i<j} \operatorname{sgn}(i-j) \operatorname{sgn}(R_i - R_j), \tag{3.11}$$

where sgn denotes the signum function that extracts the sign of a real number.

Intuitively, the Kendall's $\tau$ between two vectors will be close to 1 when their values have a similar order, and closer to 0 when values have a dissimilar rank between the two vectors. By definition, it is possible to obtain a value of $-1$ when the vectors have fully different rankings. It is possible to derive a distance measure associated with the coefficient.

**Definition 3.2 (Kendall's tau distance (Kendall, 1938)).** Let $(X_i, Y_i)$, $i = 1 \dots n$, be independent and identically distributed (iid) random vectors. Further, let $(i, R_i)$, $, i = 1 \dots n$, be paired rankings, where $R_i$ is the rank of $Y$ whose corresponding $X$ has rank $i$ (ties are neglected for simplicity). A Kendall's tau distance is defined as

$$d_\tau(X,Y) = \frac{n(n-1)(1-\tau(X,Y))}{4}, \tag{3.12}$$

which can be normalised to $[0,1]$ with

$$d_{\tau_n}(X,Y) = \frac{d_\tau(X,Y)}{n(n-1)}. \tag{3.13}$$

Normalized kendall's tau distance can be applied to provide evidence for a substantial explanation change in a sense of ranking by substituting $X := a, Y := b, n := |V|$, so we did in previous work.[6] For example in Figure 1.1, we report $d_{\tau_n} = 0.2$ to indicate a noticeable change in the variables' ranking, e.g. the sex variable was 2nd least important and becomes the 2nd most important by SHAP values.

However, there is a room for improvement. We posit that, in practice, changes in the ranking of top most important variables are far more detrimental than changes in the ranking of unimportant variables. Let's consider an explanation consisting of 20 variables for example. If a SHAP value of the most important variable (based on the original explanation) decreases to the point it becomes 4th important, it is more probable to change the human's interpretation than in the case when the least important variable moves to be the 16th important. Even in nowadays software, it is an often a default setting to only visualize top-10 attributions in an explanation, as humans can easily comprehend top-3 or top-4 values, but presenting more leads to information overload; a phenomenon well studied by Poursabzi-Sangdeh et al. (2021). We can add weights to the equation to take this behaviour into account.

[6] In (Baniecki & Biecek, 2022), we reported the *distance* value equal to 0.6, which in fact was the value of the *coefficient*. We hope to clarify this error here, which in fact does not change the overall conclusion.

**Definition 3.3 (Weighted Kendall rank correlation coefficient (Shieh, 1998)).** Let $(X_i,\ Y_i),\ i = 1 \dots n$, be independent and identically distributed (iid) random vectors. Further, let $(i,\ R_i),\ ,i = 1 \dots n$, be paired rankings, where $R_i$ is the rank of $Y$ whose corresponding $X$ has rank $i$. Let $w(i, j)$ be a weight function which is bounded and symmetric and $w : N^2 \longrightarrow R$. For simplicity, we use $w_{i,j}$ to denote $w(i, j)$. A weighted Kendall's tau is defined as

$$\tau_w(X, Y) = \frac{1}{\sum_{i,j} w_{i,j} - \sum_{i,i} w_{i,i}} \sum_{i \neq j} w_{i,j} \operatorname{sgn}(i - j) \operatorname{sgn}(R_i - R_j), \tag{3.14}$$

where sgn denotes the signum function that extracts the sign of a real number.

Overall, various weighted rank distance measures, e.g. based on weighted Kendall's tau coefficient, are proposed in the literature to account for top-$k$ importance (Piek & Petrov, 2021). We want the desired measure to use weights and have values constrained between 0 (alike ranking of SHAP values) and 1 (distinctly different explanation), similarly to the definition of $ME$. Therefore, we propose to measure ranking manipulation effectiveness defined as

$$RME(a, b) = \frac{(1 - \tau_w(a, b))}{2}, \tag{3.15}$$

using $w_{i,j} = \frac{1}{i} + \frac{1}{j}$, which is based on the `scipy.stats.weightedtau` implementation available in the `scipy` Python package (Virtanen et al., 2020).

**Evaluating explanations.** We note that studying relationships between all the previously mentioned measures in the context of various tasks (models, datasets) is considered as evaluating explanations (see Chapter 2 for similar works). Yet, apart from general guidelines concerning the use or misuse of explanations, it is useful to indicate those vulnerabilities with a specific evaluation measure for each particular explanation. Thus, we propose to measure the correlation between the change in SHAP values and baseline data. For that, we need to employ a distance measure of change in data distribution. We propose to use the Wasserstein distance (Vaserstein, 1969; Ramdas et al., 2017), also known as earth mover's distance (Levina & Bickel, 2001), which was also previously used in explainable machine learning (see Chapter 2). Compared to other measures, e.g. the Jensen-Shannon divergence (Endres & Schindelin, 2003) used by us previously (Baniecki & Biecek, 2022), Wasserstein distance does not require the distributions of variables to be defined on the same probability space, which is highly practical.

**Definition 3.4 (Wasserstein distance (Vaserstein, 1969)).** Wasserstein distance between the two distributions $u$ and $v$ is defined as

$$d_w(u, v) = \inf_{\pi \in \Gamma(u,v)} \int_{\mathbb{R} \times \mathbb{R}} |x - y| d\pi(x, y), \tag{3.16}$$

where $\Gamma(u, v)$ is the set of (probability) distributions on $\mathbb{R} \times \mathbb{R}$ whose marginal distributions are $u$ and $v$ on the first and second factors respectively.

Note that in estimation, inf becomes min, $\int$ becomes $\sum$, and Wasserstein distance becomes the distance between values of two cumulative distribution functions (Ramdas et al., 2017). For computation, we use the `scipy.stats.wasserstein_distance` implementation of the method (Virtanen et al., 2020). Since estimating joint distribution of over 10 variables presents computational challenges and is prone to high estimation error, we introduce the following aggregation procedure to measure the normalized Wasserstein distance between the two datasets.

**Definition 3.5 (Normalized Wasserstein distance).** Normalized Wasserstein distance between the two datasets $X$ and $X'$ is defined as

$$d_{W_n}(X, X') = \frac{1}{|V|} \sum_{j=1}^{|V|} \frac{d_w(X_j, {X'}_j)}{X_{j,\ min} - X_{j,\ max}}, \tag{3.17}$$

where $X_{j,\ min}$ and $X_{j,\ max}$ are minimum and maximum values of variable $X_j$, respectively.

We assume $X_{j,\ min} - X_{j,\ max} \approx X'_{j,\ min} - X'_{j,\ max}$ for simplicity.[7] The denominator $X_{j,\ min} - X_{j,\ max}$ aims to normalize the weight of each variable change, as $d_w \in [0, \infty)$ by definition. Intuitively, $\frac{d_w(X_j, {X'}_j)}{X_{j,\ min} - X_{j,\ max}} \geq 1$ is when the distributions have disjoint domains, which is an extreme case. Furthermore, we note that the choice of average as an aggregation measure is arbitrary, and it might be also useful to consider its $d^k_{W_n}$ variation of measuring only the change in top-$k$ most important variables.

Finally, we introduce a heuristic measure of explanation performance based on the following desired property connecting $ME$ & $RME$ with $d_{W_n}$. We propose that a *highly-performant* explanation instance is the one, for which we can *predict* the relationship between its change and the change in the reference data distribution. Consecutively, a *low-performant* explanation instance behaves *unpredictable*: (1) it can be arbitrarily changed without increasingly changing the reference data distribution, or (2) the reference data distribution can be arbitrarily changed without increasingly changing the explanation. This predictive performance can be measured with a simple linear regression, although the relationship does not have to be linear (we further elaborate on this in Chapter 5).

**Definition 3.6 (Explanation performance).** Let us denote by $a$ the vector representing the original explanation estimated from the baseline dataset $X_a$. Further, let $(B_i, X_i), i = 1 \dots n$, be outcomes of the manipulation method, where $B_i$ is the manipulated explanation created by substituting $X_a$ with $X_i$. We first compute vectors $\Delta B_i \coloneqq AP(a, B_i)$ and $\Delta X_i \coloneqq d_{W_n}(X_a, X_i)$ for $i = 1 \dots n$. Then, fit a linear model

$$\Delta B_i \approx \hat{\beta}_0 + \hat{\beta}_X \Delta X_i, \tag{3.18}$$

where $R^2$ and $\hat{\beta}_X$ of the fitted model becomes explanation performance denoted as $EP(a)$.

Analyzing $\hat{\beta}_X$ is worth considering to gain further context about the relationship between $B_i$ and $X_i$. Note that $RME$ can be used instead of $ME$ in the definition of explanation performance, which shows the generality of the approach. It is possible to iteratively create outcomes $(B_i, X_i), i = 1 \dots n$, using a manipulation algorithm, which we now introduce.

[7] Specifically in our case, $X_{j,\ min} - X_{j,\ max} \geq X'_{j,\ min} - X'_{j,\ max}$ by the manipulation algorithm's construction described in Section 3.5.

## 3.5. Manipulation algorithm

In this section, we describe the algorithm used for manipulating SHAP, which is an extension of our previous work (Baniecki & Biecek, 2022); thus, we conclude with highlighting the changes and improvements. Later, we use the method for evaluating the vulnerabilities of explanations in various settings (Chapter 4).

Genetic algorithms are highly versatile optimization tools (Wright, 1991; Grefenstette, 1993). Algorithm 1 presents the general idea of such an evolutionary strategy. First, a population of individuals is created. Then, in each iteration, these individuals cross features with each other to create new individuals. These are then mutated at random with a given probability using a predefined set of possible changes. Each individual is evaluated with a fitness function indicating the optimization criteria. Finally, a ranking and selection method is used to reduce the population to the original number of individuals for the next iteration. Repeating the process converges to a given set of results, where the best individual is an optimum result. Overall, a versatile formulation of the genetic algorithm has several advantages: (1) the fitness function might be of any kind, without restrictions on its differentiability, (2) there is a high exploration potential in considering various crossover and mutation operators, and (3) one can find the best optimization result given a predefined resource budget, e.g. time, computational resources.

**ALGORITHM 1:** Genetic algorithm

```
initialize the population of individuals
for i = 1...max_iter do
    crossover: enlarge the population by combining individuals into new ones
    mutation: modify individuals with a predefined set of possible (random) changes
    evaluation: compute the phenotype from genotype, and then fitness values
    if i $\neq$ max_iter then
        selection: reduce the population based on a ranking of fitness
    end
end
select the best individual
```

**Setting.** Let us denote by $a$ the vector representing the original explanation estimated from the baseline dataset $X_a$. We use the explanation $b$ as a phenotype of an individual, whose genotype (also called chromosomes) is the corresponding validation set instance $X_b$ used as a background dataset for computing the explanation. The objective is to optimize the following fitness function

$$\mathcal{F}(X_b) = \alpha \cdot d_1(a,b) + (1-\alpha) \cdot d_{\tau_n}(a,b) - \beta \cdot d_{W_n}(X_a, X_b), \tag{3.19}$$

where one aims to find $X_b$, which maximizes $d_1$ and $d_{\tau_n}$, while minimizing $d_{W_n}$. Parameter $\alpha$ is responsible for the tradeoff between changing the original explanation's values and ranking. Parameter $\beta$ serves as a regularization; specifically, $\beta = 0$ removes the constraint of aiming at minimal change in data distribution. Although conventionally genetic algorithms aim to *maximize* the fitness function, we modify it and consider *minimizing* the following loss function

$$\mathcal{L}(X_b) = \alpha \cdot d_1(t,b) + (1-\alpha) \cdot d_{\tau_n}(t,b) + \beta \cdot d_{W_n}(X_a, X_b), \tag{3.20}$$

where $t$ is an arbitrary predefined target vector. This greatly improves the conditioning of the optimization problem in practice. However, one has to define $t$ for each problem separately.

Examples include $t \coloneqq mean(a)$, $t \coloneqq median(a)$, and $t \coloneqq reverse(a)$, depending on the context. We found $t \coloneqq mean(a)$ to be quite successful, because the distribution of $a_j$ values is usually right-skewed, and $t \coloneqq median(a)$ becomes too small for meaningful results.

**Operators.** We considered various mutation, crossover, and selection operators for a reasonable framework. The least specific is **selection** algorithm, which chooses $m$ individuals, each with a probability proportional to its ranking based on the loss values. We propose the probability $p_i = \frac{2i}{m(m+1)}$ assuming the fittest individual has rank $m$, and the least fit individual has rank 1. This way $\sum_{i=1}^{m} p_i = 1$. We apply a natural approach of sampling individuals with replacement. Additionally, we use a simple *elitism* mechanism to improve convergence by making sure that no best solution is lost between iterations, e.g. due to too aggressive exploration. It means that always $k$ number of best individuals survive to the next iteration, which *does not* exclude them from being sampled again during the selection process. Next, for the **crossover** operator, we considered swapping rows or columns between the two individuals (datasets). The first can substantially change the distributions, while the second involves encoding little new information in the process. Thus, we decide a intermediate approach by swapping randomly half of values between the two parent individuals, while acknowledging that this proportion could be further parameterized. Finally, the **mutation** operator becomes the most specific based on the considered data type and values. We mutate each individual with a given parameterized probability and apply to it the following heuristics: (1) Numerical variables are changed with a given probability by adding a Gaussian noise generated with zero mean and sigma proportional to the standard deviation of each variable. (2) Such mutation is constrained to the domain of the original variable's values by updating values exceeding the maximum or minimum. (3) Categorical variables are changed with a given probability by sampling values from the original variable's domain (set of possible values). We note that some observations might evolve out-of-distribution in the process, which is not necessarily a limitation, more so a factor to be considered when interpreting the result. Thus, we consider the Wasserstein distance a more adequate measure of distribution change than Jensen-Shannon. We efficiently implement the above mentioned operators using $n$-dimensional arrays available in the `numpy` Python package (Harris et al., 2020).

**Parameters.** Table 3.2 reports the default values of the algorithm's parameters, which we estimated empirically, and set constant during all the experiments. We believe that tuning these parameters to improve the convergence is out of scope of this work, and we discuss this possibility as future work in Chapter 5.

**Major changes with respect to (Baniecki & Biecek, 2022).** A few modifications to the previous version of the genetic algorithm are worth highlighting here. Firstly, we improved the loss function (and evaluation measures) to better reflect an idea of maximizing absolute and ranking distances while regularizing the optimization with a distance between data distributions. Secondly, we implemented a mutation operator for categorical variables. Third is a modification of the crossover column wise operator to balance the tradeoff between the column wise and row wise possibilities of exchanging parent values. Overall, the implementation of the algorithm is more efficient (in a computational sense) due to improving the calculation of explanations. We now provide support for `torch` neural network models explained using the GradientSHAP attribution from `captum`.

Table 3.2: Description of the manipulation algorithm's parameters.

| **Name** | **Default** | **Description** |
|---|---:|---|
| `pop_count` | 51 | Number of individuals in the population. |
| `top_survivors` | 2 | Number of top individuals surviving to the next iteration through the elitism mechanism. |
| `mutation_prob` | 0.5 | Probability of mutation for each individual. |
| `std_ratio` | $\frac{1}{9}$ | Fraction to multiply the original standard deviation in generating noise for a given variable. |
| `crossover_ratio` | 0.5 | Fraction of individuals that take part in the crossover in a given iteration. |
| `alpha` | 0.9 | Optimization parameter $\alpha$. |
| `beta` | 0.1 | Regularization parameter $\beta$. |
| `max_iter` | 300 | Number of iterations (epochs). |
| `stop_iter` | 20 | Number of iterations needed to stop the algorithm due to breaking the early stopping condition (patience). |
| `epsilon` | 1e$-$3 | Stopping parameter $\epsilon$. |

# 4. Results

In this chapter, we report results from experiments conducted using the described methods. Section 4.1 is to provide an exemplary visual path from manipulating to evaluating explanations of machine learning models. Section 4.2 provides more details on this process. Finally, we benchmark explanations in various data and model settings in Section 4.3.

## 4.1. Illustrative example

We consider the `small` variant of the `gaussian` dataset, which is a binary classification task, and fit a feedforward neural network with 2 hidden ReLU layers of 32 neurons (`FNN:2:32`) achieving a performance of 0.88 $AUC$ and 0.80 $F_1$. Next, we use GradientSHAP to explain predictions for $K = 10$ observations of interest, depicted with blue in Figure 4.1. To evaluate the explanation, we run the manipulation algorithm 3 times for 300 iterations each. Figure 4.2 presents convergence of the best run achieving a performance of 0.54 $ME$ and 0.17 $RME$. This indicates a substantial change in both absolute values of the explanation, as well as its ranking, depicted with red in Figure 4.1. Specifically, the importance of $X_3$ is reduced by about 60% and $X_6$ becomes top-6 most important variable. For context, Figure 4.3 shows a distribution shift occurring in the dataset, for which the normalized distance $d_{W_n}$ between the distributions equals 0.054.

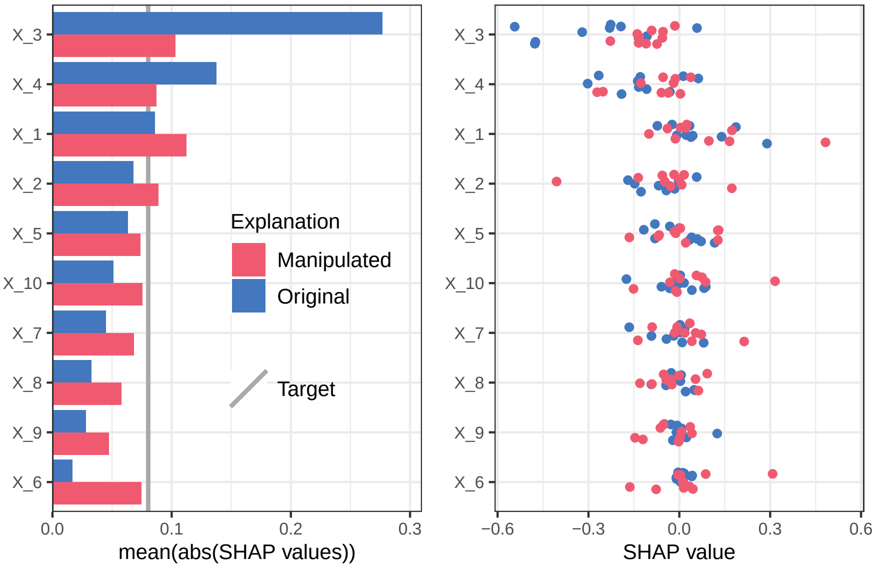

Figure 4.1: Comparison between the original explanation and the one manipulated using the genetic algorithm (`gaussian_small:FNN:2:32`). **Left:** Aggregated SHAP importance explanation. **Right:** SHAP attributions for 10 observations (points per row).

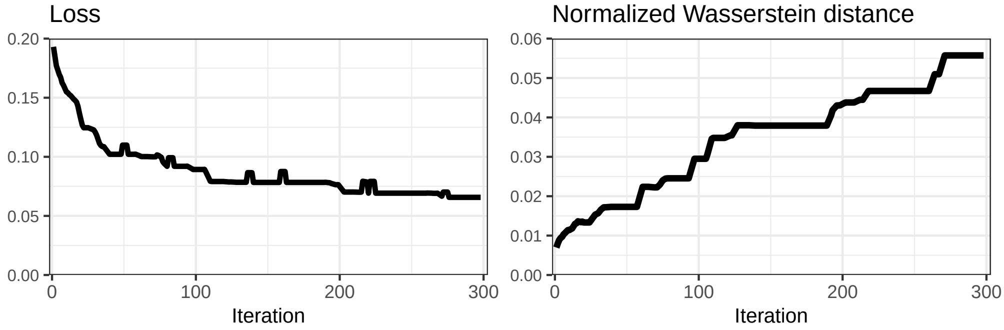

Figure 4.2: Convergence of manipulating a GradientSHAP explanation of the `FNN:2:32` model fitted to the `gaussian_small` dataset.

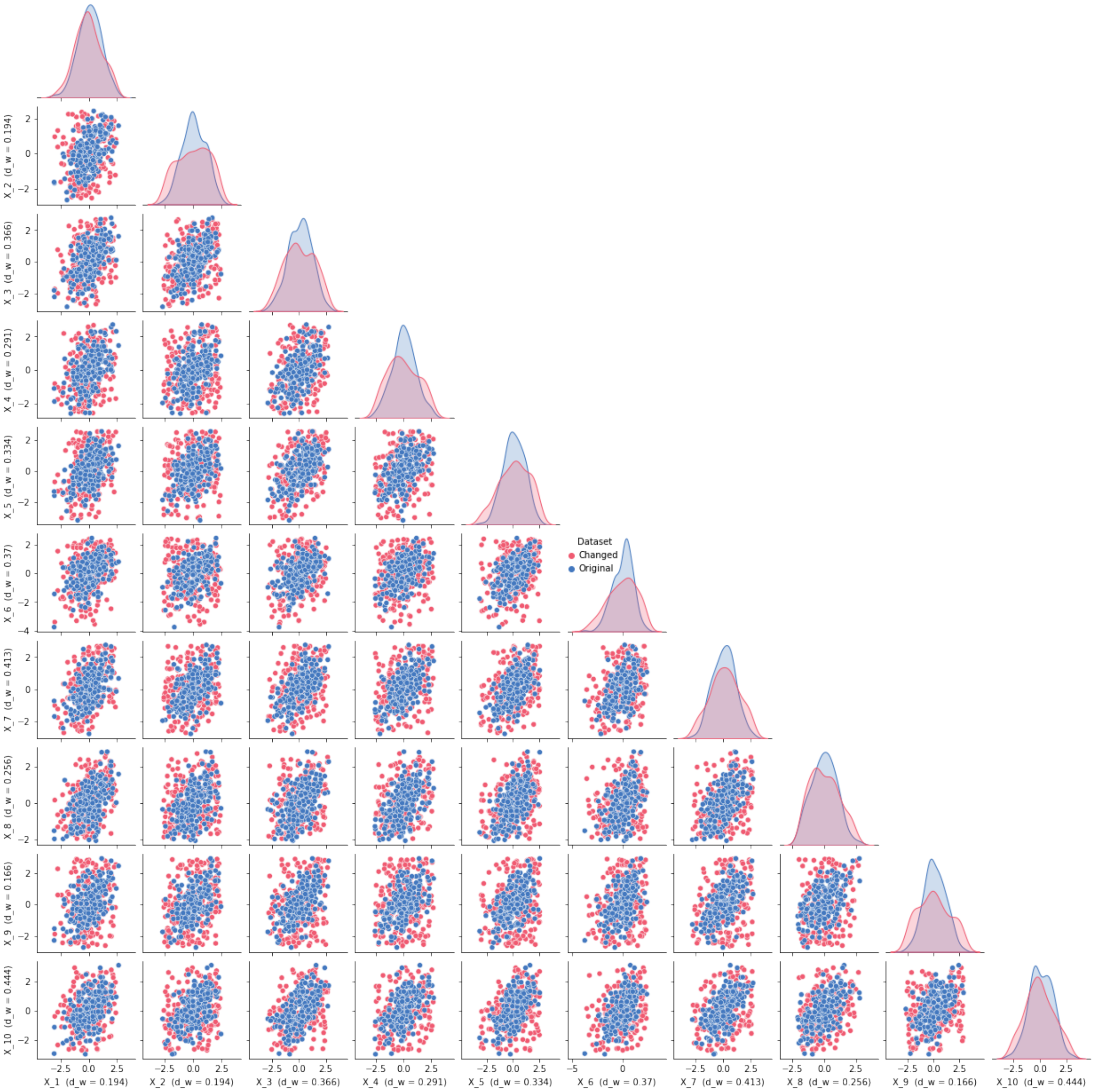

Figure 4.3: Comparison between the original `gaussian_small` dataset and the one changed using the manipulation algorithm. For broader context, we report the Wasserstein distance $d_w$ for each variable next to its axis.

What is the takeaway from this example? Both the explanation and baseline distribution changed in the process of manipulation. However, it is useful to measure the explanation performance (see Definition 3.6), which in this case equals 0.74 $R^2$ and $\hat{\beta}_X = 1.92$. These values can lead to the conclusion that the explanation in question behaves as predicted, meaning that increasing the shift in data heavily increases its change.

## 4.2. Instructive case

To illustrate a real-world application, we provide an instructive case to highlight a potential danger in model audit. Let's consider a healthcare machine learning task of predicting an Ischaemic heart disease in hospitalized patients, which is a binary classification task with 10 variables (`heart_small`). We fit a gradient boosting decision tree with 100 trees and maximal depth of 4 achieving a performance of 0.82 $AUC$ and 0.65 $F_1$ (`GBDT:4:100`). TreeSHAP can be used to explain the predictions for patients of interest, depicted with blue in Figure 4.4. There are two sides to applying our manipulation algorithm: (**A**) an *adversary* may look to deceive an *auditor* by swapping the background dataset used for explanation computation, (**B**) an *auditor* may want to evaluate if and how shifts in the background dataset change the explanation in question. Such shifts may occur for various reasons, e.g. change in validation split or out-of-time drift.

Figure 4.5 presents convergence of the best run achieving 0.42 $ME$ and 0.14 $RME$, which indicates a visible change in explanation colored with red in Figure 4.4. For example, the importance of variable Age is reduced by about 75% and variable Gender becomes more important than before. Note a substantial change in the sign of SHAP values for variable Maximal PTT (right subplot). For side **A**, it looks like a successful manipulation, while for side **B** it might raise concerns about the quality of original SHAP attribution values.

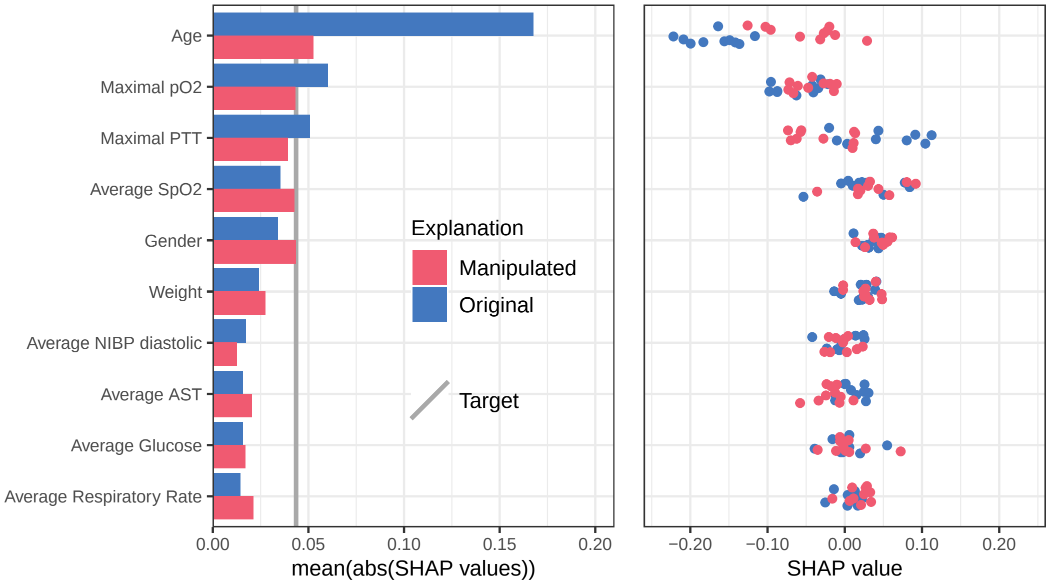

Figure 4.4: Comparison between the original explanation and the one manipulated using the genetic algorithm (`heart_small:GBDT:4:100`). **Left:** Aggregated SHAP importance explanation. **Right:** SHAP attributions for 10 patients (points per row).

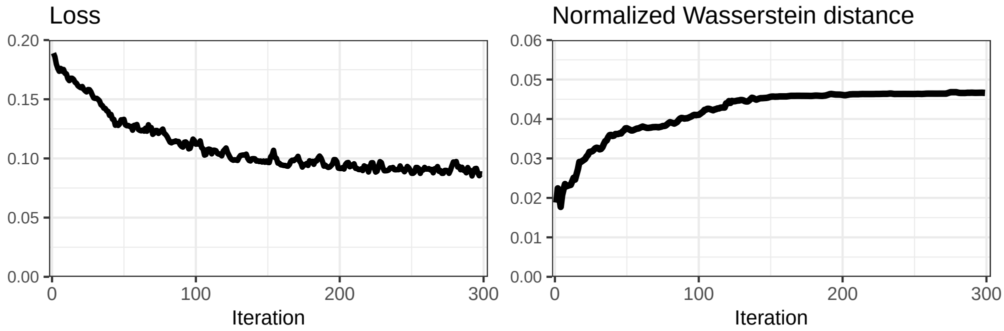

Figure 4.5: Convergence of manipulating a TreeSHAP explanation of the `GBDT:4:100` model fitted to the `heart_small` dataset.

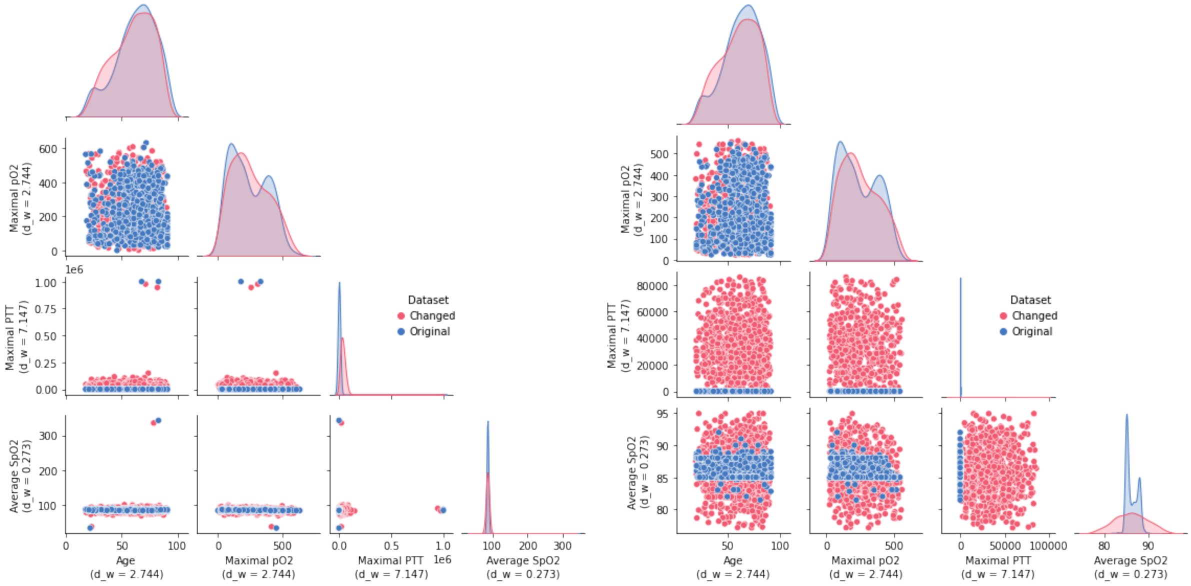

Figure 4.6: Differences in four most important variables between the original `heart_small` dataset and the one changed using the manipulation algorithm. For broader context, we report the Wasserstein distance $d_w$ for each variable next to its axis. **Left:** Distribution of four variables. **Right:** Distribution of the same data with about 5% outlier observations removed.

The context of reference data distribution is crucial in interpreting model explanations. Figure 4.6 shows a distribution shift among four most important variables in the dataset, for which the normalized distance $d_{W_n}$ calculated on all variables equals 0.046. The explanation performance equals 0.94 $R^2$ and $\hat{\beta}_X = 3.51$. We present two subplots differing in the number of observations (points), i.e. the right subplot has about 5% outlier observations removed by clipping the variables' values to the range $[quantile_{1\%}, quantile_{99\%}]$. Notice that the shift appears to be small based on the original data, and looks substantial when zooming in on the majority of data distribution mass. Alarmingly, this change in data becomes undetectable by Normalized Wasserstein distance due to the variable's range appearing in the measure's denominator. This example puts to debate the ability of auditor to find out an adversary in scenario **A**. What if there are more than 10 variables? For example, in the `large` tasks, we consider 100 explanatory variables, for which it is infeasible to visualize all distributions to visually evaluate the context of explanation. Hence, a robust measure quantifying change in data is needed, also to support the auditor in scenario **B**. We further discuss alternative normalization procedures in Chapter 5.

## 4.3. Benchmark

Chapter 3 mentions 12 dataset tasks and 32 model configurations, which results in 384 predictive tasks consisting of a dataset and model pair. We run the manipulation algorithm with default parameters 3 times for most of the tasks, and 2 times for computationally demanding configurations, e.g. a large dataset and model, which results in about 1000 experiments in total. This is to minimize randomness by choosing the best out of 2-3 algorithm runs based on the loss value for the final analysis. The benchmark took about 10 days to compute on a server node consisting of two AMD EPYC 7413 CPUs and 256GB RAM allowing us to run the experiments in 80 threads in parallel.

As mentioned in Section 3.3, the target is a semi-global explanation for $K = 10$ observations chosen for each predictive task separately. We report performance measures of all the models created for the purpose of this study in Tables 4.1, 4.2, 4.3 & 4.4 at the end of the Chapter, which is to provide additional background for the analysis of manipulating and evaluating explanations. In general, we aim to evaluate explanations with respect to: (1) model size and type, which is unequivocally bound to the type of explanation, and (2) dataset size and type.

**Comparison between models.** Figure 4.8 presents distribution of the considered performance measures for all the best algorithm runs (384). We clearly observe that, in general, GradientSHAP explanations of FNNs are far more vulnerable to the attacks than TreeSHAP explanations of GBDTs. There is a higher variance of both explanation and data change in the case of FNNs. Figure 4.8 extends these results by reporting two dimensional relationships among the experiments for each model type. There is little evidence that differences in the vulnerability of FNN and GBDT come from either substantial change in data or the varied complexity of models. By model complexity, we specifically mean the number of parameters, which is calculated as $\texttt{n\_trees} \cdot 2^{(\texttt{max\_depth}-1)}$ for GBDTs and $\texttt{layer\_count} \cdot (\texttt{input\_dim} + \texttt{output\_dim}) + (\texttt{layer\_count} - 1) \cdot \texttt{neuron\_count}^2$ for FNNs. We use $(\texttt{max\_depth} - 1)$ instead of `max_depth` because the latter effectively provides an upper bound on the number of leaves, which is not the case in practice ($\texttt{max\_depth} - n$ could be considered).

Figure 4.9 shows an in-depth analysis of the hyperparameters' effect on the change in explanations and data. To quantify the differences in populations, we perform Analysis of Variance (ANOVA) using the `Anova(lm(me∼max_depth*n_trees), type=3)` function available in the `car` R package (Fox & Weisberg, 2019) taking into account both hyperparameters in the analysis with their interaction:

- For GBDT, we fail to reject the F-Test's null hypothesis while noticing no unusual relationships between the manipulation effectiveness against: maximal depth (p-value = 0.690), number of trees (p-value = 0.953), or interaction between the two (p-value = 0.928).
- Consecutively for FNN, there is no such relationship between the manipulation effectiveness against: number of neurons (p-value = 0.988), number of layers (p-value = 0.118), interaction between the two hyperparameters (p-value = 0.806).

We note that removing interaction from the equation provides evidence for the number of layers being significant (p-value = 0.019), which is indeed observable in Figure 4.9. The results for $RME$ are consistent with those for $ME$.

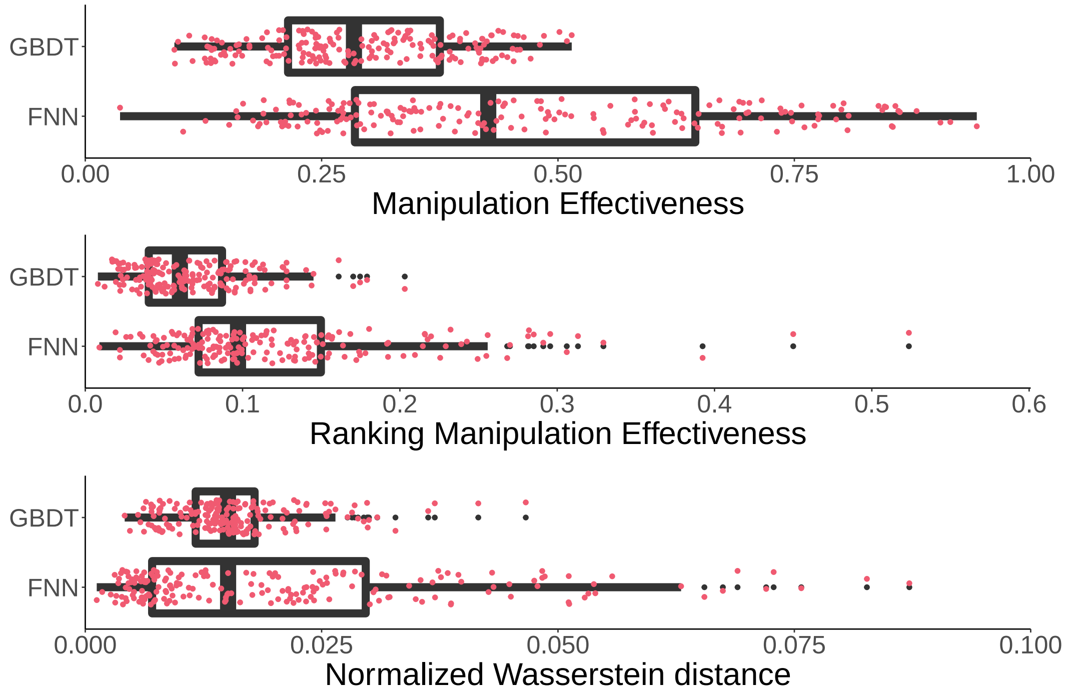

Figure 4.7: Distribution of performance measures for all the considered experiments.

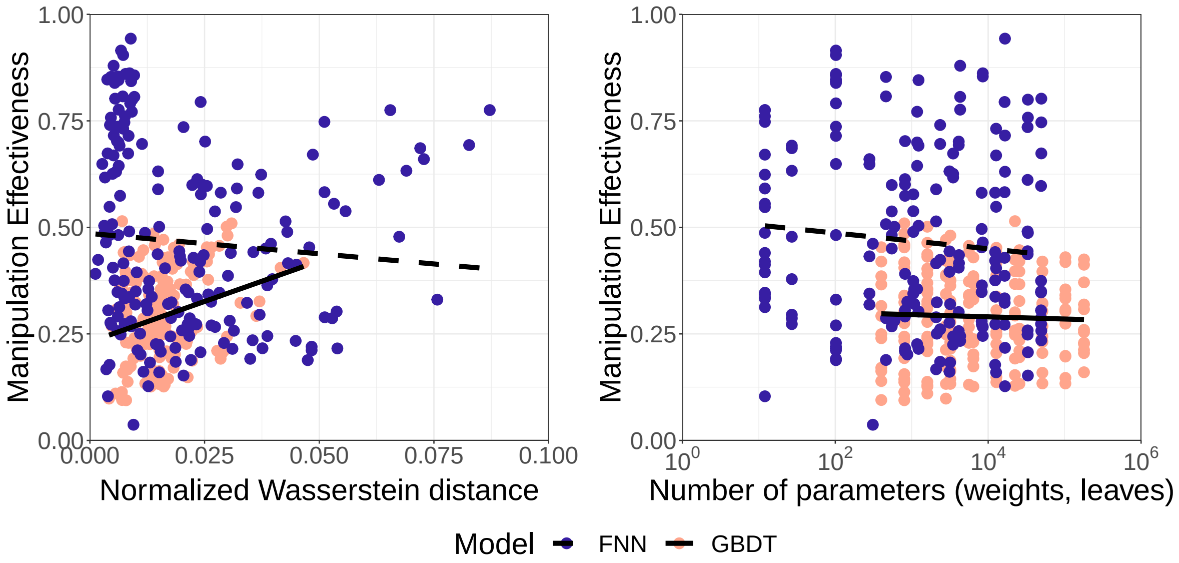

Figure 4.8: Relationship between manipulation effectiveness, data distribution change, and the number of model's parameters for all the considered experiments.

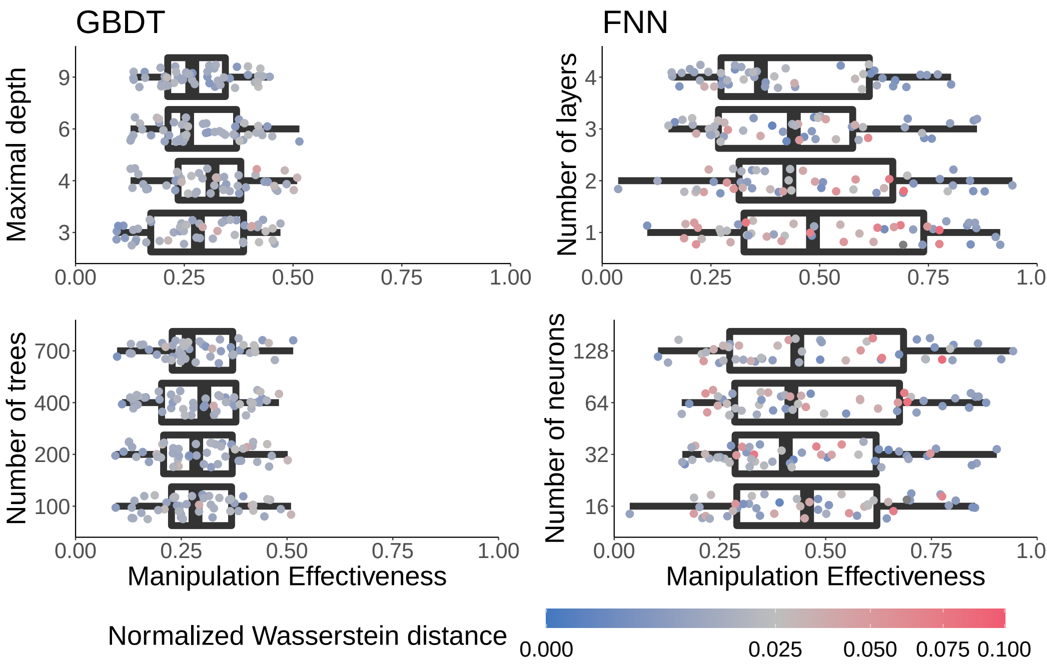

Figure 4.9: Relationship between the models' parameters and the change in explanations and data distribution.

**Comparison between datasets.** As mentioned in Section 3.1, `mimic` validation datasets consist of 1000 observations and either 10 or 100 variables, and `xaibench` datasets consist of 200 observations and either 10, 25, or 100 variables. Figure 4.10 shows distributions of manipulation effectiveness per each dataset type and size. It is again observable that explanations of FNNs are more vulnerable. Variances of manipulation effectiveness values are quite low between different GBDTs in each task. There are far lower differences in $RME$ than in $ME$ across all the datasets, as we mainly optimized the manipulation algorithm to maximize $ME$ ($\alpha = 0.9$). We perform ANOVA using `Anova(lm(me~dataset_size+dataset_task), type=3)`, this time without interactions due to non-existing value combinations like `heart_medium`, to conclude the following:

- For GBDT, we reject the null hypothesis proving the relationship between $ME$ versus dataset size (p-value $< 0.001$) and task (p-value $< 0.001$).
- For FNN, notice a relationship between $ME$ versus dataset task (p-value $< 0.001$), while the effect is unclear for dataset size (p-value $= 0.038$).

The results for $RME$ are consistent. In fact, the trend for $ME$ is visible in Figure 4.10, where for at least 3 out of 5 datasets the mean manipulation effectiveness is smaller on large datasets than on small datasets. Please note that, with more variables (in our case $10 \ll 100$) the SHAP values become naturally smaller (they sum up to the prediction) and their mean aggregation in $ME$ becomes a larger approximation of the whole phenomenon. Therefore, the interpretation of $ME$ might change with respect to the number of variables in the dataset and the same case can be made for $d_{W_n}$. Nevertheless, there is a clear distinction between GBDTs and FNNs in the above analysis.

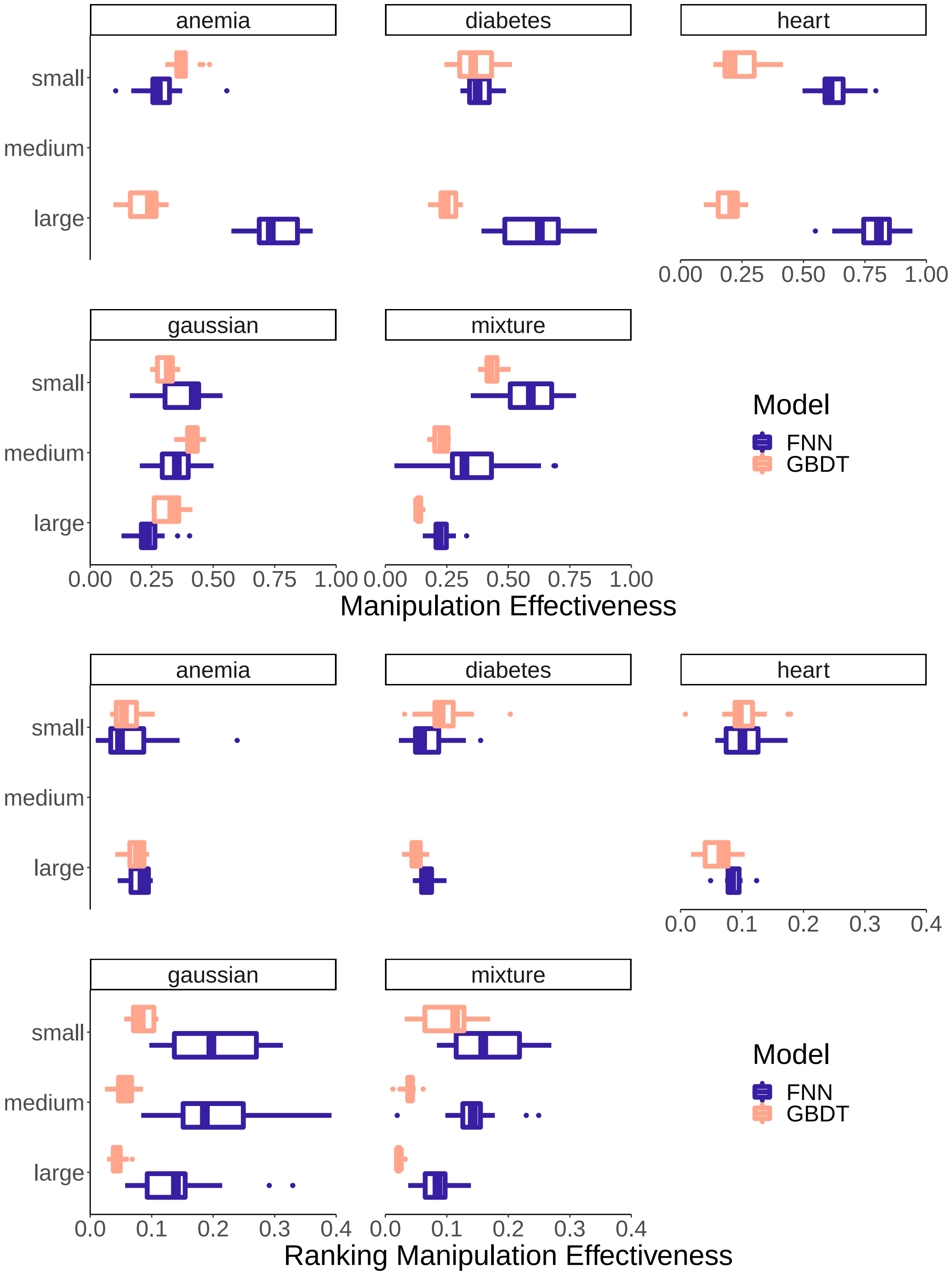

Figure 4.10: Relationship between the data parameters and manipulation effectiveness. Only for visualization clarity, we removed two outliers in `gaussian_small` that had over 0.4 $RME$, which is visible in Figure 4.7.

**Global analysis.** To summarise the benchmark, we propose a global analysis approach. We first fitted a linear regression model to quantify all the abovementioned factors relating to the manipulation effectiveness. However, there are many interactions between these explanatory variables by design, which made it difficult to interpret the result. For example, from data analysis we know that there models behave differently between tasks (Figure 4.10). Model type and its number of parameters correlates with performance (Tables 4.1-4.4). Also, some nuances like the fact that there is no `medium` size of `mimic` datasets make it harder to perform one-hot encoding of categorical experiment parameters. Summing up, it is challenging to analyze this benchmark with a linear model to infer about the significance of its covariates.

Therefore, we propose to use the decision tree algorithm for global analysis, as it better captures interactions between the variables. Figure 4.11 shows a visualization of decision trees for the `mimic` tasks – we develop separate trees as oppose to adding another variable (dataset name) to the equation to remove the potential factor of: the choice of task indeed affects the normalization of $ME$. We analyze variable splits in trees to infer the following:

1. The most important factor to the analysis is model type, which we believe directly corresponds to the differences between TreeSHAP and GradientSHAP. Note that a split by model type is not evident in the `anemia` task, but the first split of $F_1 < 65$ can be viewed as such. This is because Table 4.2 reports that no GBDT model achieves more than 64.4 $F_1$ in this scenario and most FNN models achieve more than 64.9 $F_1$.

2. The change in reference data distribution (Normalized Wasserstein distance) appears to be strongly aligned with $ME$. It accounts for $n = 10,\ 38\%$ splits in all the three trees, where in $n = 9,\ 90\%$ of those splits it is a positive association, i.e. if $d_{W_n}$ increases then $ME$ increases.

3. The number of models' parameters (weights, leaves) may be a potential predictor of $ME$ as it accounts for $n = 6,\ 23\%$ splits in total. The results seem ambiguous as we observe a positive association in $n = 4,\ 67\%$ out of these splits.

4. Consequently, the model's performance ($F_1$) becomes important with $n = 6,\ 23\%$ splits (omitting the one correlating with model type). The association with $ME$ is also ambiguous as $n = 3,\ 50\%$ splits manifest an increasing relationship.

5. There is only one split that uses the dataset size (number of variables) across the three predictive tasks (`diabetes:FNN`). At first glance, this contradict the significance obtained with ANOVA, which might be due to unaccounted interactions.

Knowing this, we want to highlight a large difference in the decision tree fitted for the `gaussian` task, which is presented in Figure 4.12. It gives no importance to model type and proves strong association between the dataset size and $ME$. Findings about change in data, number of parameters and performance are similar to those for `mimic`. A takeaway from this example is that the explanations' vulnerabilities may greatly vary between different scenarios, but even more between synthetic data and real-world applications. Thus, evaluating explanations in many contexts is beneficial for the overall understanding of their limitations.

Stakeholders could analyze such tree-like visualizations to *globally* better interpret the explanations' results in a given domain, accounting for the type of model, its complexity and performance. We here present such application on an example of medical data.

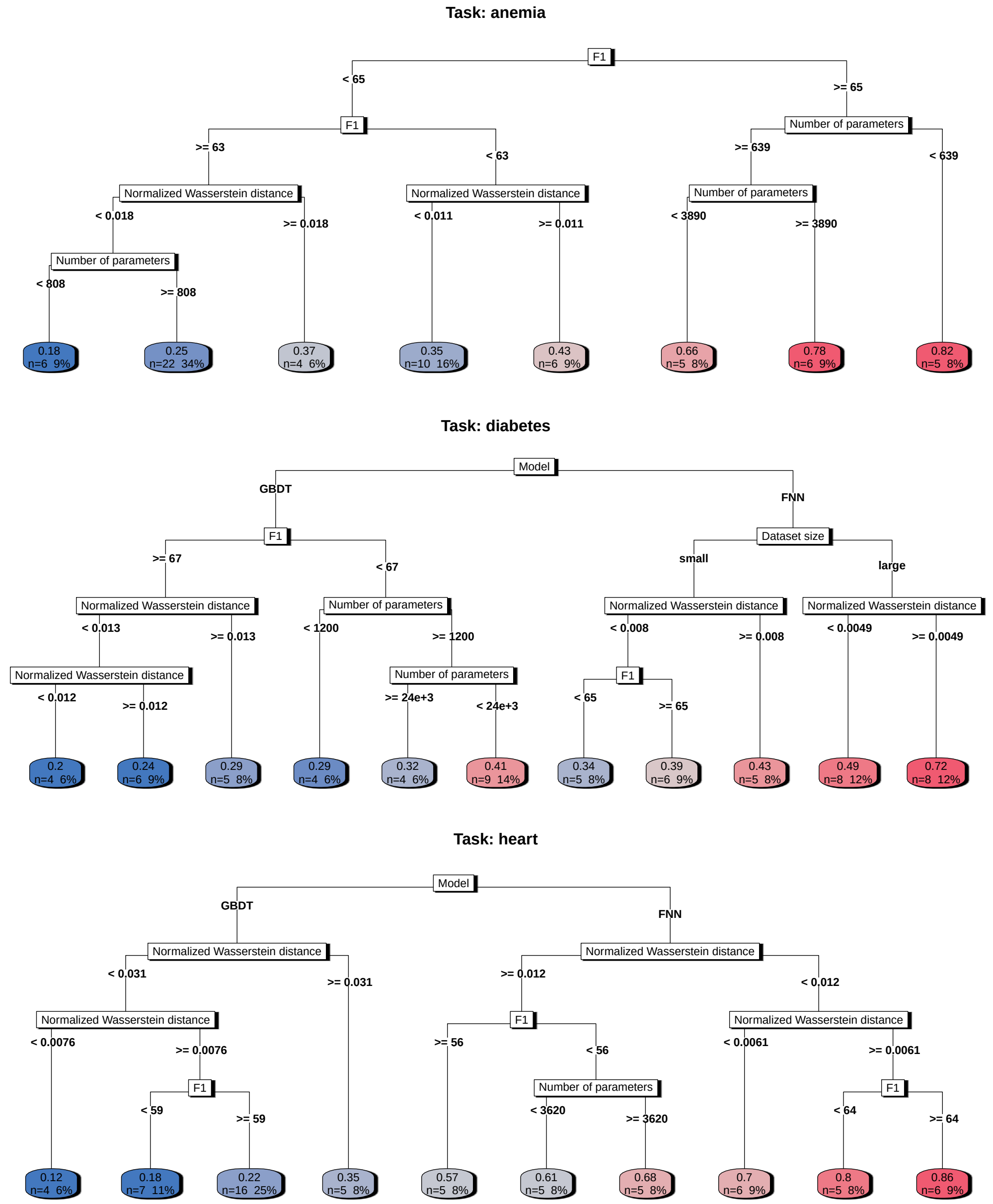

Figure 4.11: Decision trees predicting manipulation effectiveness based on the parameters and results of experiments on the `mimic` tasks.

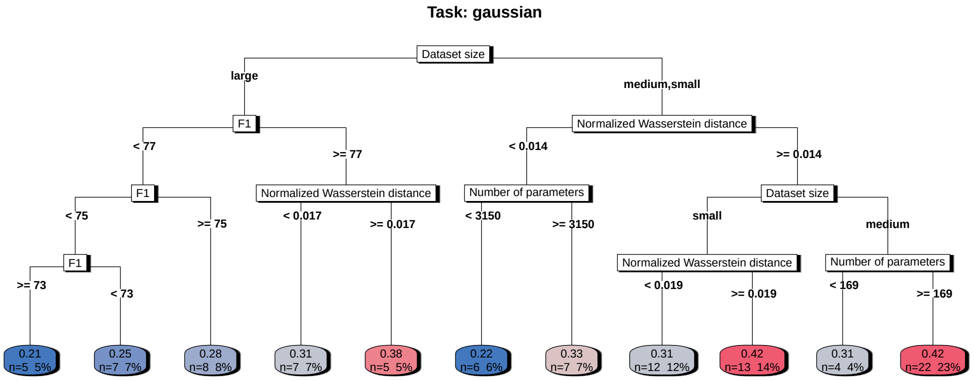

Figure 4.12: Decision tree predicting manipulation effectiveness based on the parameters and results of experiments on the `gaussian` task.

Table 4.1: Performance of GBDT classifiers with varied tree depth and number of trees; from the top: $F_1 \uparrow$ and $AUC \uparrow$ (scaled ×100).

| max_depth | **3** | | | | **4** | | | | **6** | | | | **9** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n_trees | **100** | **200** | **400** | **700** | **100** | **200** | **400** | **700** | **100** | **200** | **400** | **700** | **100** | **200** | **400** | **700** |
| anemia_large | 64.1 | 64.1 | 63.8 | 64.0 | 64.2 | 64.0 | 63.7 | 63.9 | 64.1 | 64.2 | 63.8 | 64.4 | 63.9 | 64.3 | 64.0 | 64.2 |
| anemia_small | 62.0 | 61.8 | 61.9 | 61.6 | 62.1 | 62.1 | 61.5 | 61.3 | 61.9 | 61.7 | 61.0 | 59.8 | 61.7 | 60.7 | 60.3 | 59.7 |
| diabetes_large | 65.9 | 67.9 | 68.7 | 68.8 | 67.2 | 68.9 | 69.1 | 68.9 | 67.8 | 68.2 | 68.0 | 68.2 | 67.6 | 67.6 | 68.3 | 68.3 |
| diabetes_small | 63.9 | 64.6 | 64.8 | 64.4 | 64.3 | 64.5 | 64.7 | 64.0 | 64.7 | 64.2 | 63.5 | 63.1 | 64.1 | 63.8 | 63.4 | 63.3 |
| heart_large | 62.6 | 64.2 | 65.1 | 65.3 | 63.6 | 64.8 | 65.9 | 65.7 | 64.7 | 65.3 | 65.3 | 65.7 | 64.2 | 64.6 | 65.4 | 65.2 |
| heart_small | 59.3 | 59.6 | 59.8 | 59.5 | 59.5 | 59.6 | 59.2 | 59.1 | 60.1 | 59.5 | 58.9 | 58.6 | 59.0 | 58.7 | 58.1 | 57.8 |
| gaussian_large | 73.1 | 76.5 | 78.3 | 78.6 | 76.8 | 77.7 | 78.0 | 77.4 | 78.3 | 79.1 | 77.1 | 78.0 | 76.6 | 77.7 | 80.3 | 82.9 |
| gaussian_medium | 82.2 | 87.4 | 83.2 | 82.7 | 77.6 | 76.0 | 77.6 | 80.0 | 76.7 | 79.7 | 82.4 | 82.1 | 80.3 | 80.8 | 80.0 | 81.3 |
| gaussian_small | 86.0 | 85.1 | 83.6 | 84.7 | 84.6 | 83.2 | 83.2 | 84.4 | 82.3 | 82.8 | 83.7 | 83.9 | 85.1 | 85.1 | 84.2 | 85.1 |
| anemia_large | 82.2 | 82.4 | 82.1 | 81.9 | 82.4 | 82.3 | 82.1 | 81.8 | 82.1 | 81.9 | 81.7 | 81.7 | 81.8 | 81.8 | 82.0 | 82.1 |
| anemia_small | 80.6 | 80.6 | 80.4 | 80.0 | 80.6 | 80.4 | 80.0 | 79.4 | 80.3 | 79.8 | 79.0 | 78.1 | 79.7 | 79.0 | 78.3 | 77.9 |
| diabetes_large | 88.3 | 88.8 | 88.9 | 88.8 | 88.7 | 88.9 | 88.8 | 88.7 | 88.7 | 88.7 | 88.7 | 88.7 | 88.6 | 88.7 | 88.9 | 89.0 |
| diabetes_small | 87.0 | 87.2 | 87.1 | 86.8 | 87.2 | 87.2 | 86.9 | 86.4 | 87.2 | 86.8 | 86.1 | 85.6 | 86.6 | 86.2 | 85.8 | 85.6 |
| heart_large | 84.1 | 84.5 | 84.6 | 84.5 | 84.4 | 84.6 | 84.6 | 84.4 | 84.6 | 84.7 | 84.6 | 84.5 | 84.4 | 84.6 | 84.8 | 84.9 |
| heart_small | 81.6 | 81.5 | 81.4 | 81.0 | 81.7 | 81.6 | 81.1 | 80.5 | 81.6 | 81.0 | 80.2 | 79.5 | 80.9 | 80.3 | 79.5 | 79.0 |
| gaussian_large | 92.2 | 93.2 | 93.7 | 93.9 | 92.4 | 93.5 | 93.9 | 94.2 | 92.0 | 92.7 | 93.3 | 93.7 | 90.5 | 91.4 | 92.7 | 93.3 |
| gaussian_medium | 94.9 | 95.3 | 94.5 | 94.1 | 93.1 | 93.4 | 93.3 | 93.3 | 92.8 | 93.9 | 94.3 | 94.0 | 93.0 | 93.5 | 93.8 | 93.5 |
| gaussian_small | 94.1 | 94.5 | 94.5 | 94.7 | 94.0 | 94.3 | 94.4 | 94.4 | 93.8 | 94.3 | 94.5 | 94.5 | 93.6 | 94.0 | 94.2 | 94.3 |

Table 4.2: Performance of FNN classifiers with varied number of layers and neurons; from the top: $F_1 \uparrow$ and $AUC \uparrow$ (scaled ×100).

| layer_count | 1 | | | | 2 | | | | 3 | | | | 4 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| neuron_count | **16** | **32** | **64** | **128** | **16** | **32** | **64** | **128** | **16** | **32** | **64** | **128** | **16** | **32** | **64** | **128** |
| anemia_large | 66.1 | 66.3 | 65.6 | 66.1 | 66.3 | 66.8 | 66.2 | 66.1 | 65.6 | 66.2 | 66.3 | 66.4 | 65.6 | 66.6 | 65.4 | 66.9 |
| anemia_small | 64.9 | 65.0 | 64.8 | 65.3 | 65.1 | 65.0 | 64.9 | 64.5 | 64.8 | 64.9 | 64.1 | 65.1 | 64.5 | 64.7 | 63.9 | 64.1 |
| diabetes_large | 67.3 | 68.1 | 67.8 | 68.6 | 67.3 | 68.0 | 67.5 | 68.0 | 68.1 | 68.0 | 68.0 | 67.9 | 66.9 | 67.1 | 67.1 | 66.6 |
| diabetes_small | 64.1 | 63.4 | 64.4 | 64.8 | 64.7 | 64.8 | 66.0 | 65.1 | 64.9 | 65.4 | 64.6 | 65.0 | 65.5 | 64.6 | 65.5 | 64.1 |
| heart_large | 65.2 | 65.3 | 64.5 | 64.6 | 64.4 | 64.6 | 64.0 | 64.8 | 63.4 | 61.9 | 64.4 | 63.8 | 64.6 | 64.8 | 64.5 | 63.4 |
| heart_small | 56.4 | 54.5 | 56.2 | 56.6 | 32.5 | 51.1 | 51.2 | 56.2 | 57.2 | 51.7 | 60.3 | 53.4 | 56.6 | 48.1 | 53.9 | 56.2 |
| gaussian_large | 72.9 | 77.0 | 76.6 | 74.8 | 74.0 | 76.1 | 74.7 | 75.2 | 73.3 | 72.8 | 70.3 | 69.9 | 70.9 | 76.1 | 75.3 | 72.7 |
| gaussian_medium | 73.3 | 74.2 | 78.7 | 75.3 | 74.0 | 75.6 | 72.8 | 78.9 | 83.0 | 77.9 | 76.7 | 74.0 | 68.5 | 75.0 | 74.2 | 75.2 |
| gaussian_small | 85.2 | 84.0 | 82.5 | 85.4 | 81.1 | 80.4 | 81.5 | 81.3 | 84.9 | 82.8 | 85.9 | 83.9 | 80.2 | 79.8 | 81.3 | 80.0 |
| anemia_large | 81.6 | 81.3 | 81.4 | 81.2 | 81.7 | 81.3 | 81.6 | 81.5 | 81.3 | 81.4 | 81.4 | 81.6 | 81.2 | 81.5 | 81.3 | 81.5 |
| anemia_small | 80.2 | 80.1 | 80.2 | 80.1 | 79.8 | 80.2 | 80.2 | 80.0 | 80.0 | 80.0 | 79.9 | 80.0 | 80.0 | 79.9 | 80.1 | 79.9 |
| diabetes_large | 87.1 | 87.0 | 86.9 | 87.2 | 87.2 | 87.1 | 87.3 | 86.9 | 86.9 | 87.0 | 87.3 | 87.0 | 87.1 | 87.2 | 86.3 | 86.6 |
| diabetes_small | 86.0 | 85.9 | 86.1 | 86.1 | 86.4 | 86.4 | 86.3 | 86.1 | 86.3 | 86.3 | 86.3 | 86.2 | 85.8 | 86.1 | 86.1 | 86.0 |
| heart_large | 82.8 | 82.7 | 82.3 | 82.0 | 82.6 | 82.9 | 82.9 | 81.8 | 83.0 | 82.8 | 82.8 | 82.7 | 82.9 | 83.0 | 82.7 | 82.2 |
| heart_small | 76.4 | 78.8 | 79.4 | 78.9 | 73.9 | 78.4 | 79.0 | 79.7 | 79.7 | 78.7 | 79.5 | 75.7 | 77.8 | 75.6 | 74.4 | 76.1 |
| gaussian_large | 86.3 | 87.1 | 87.6 | 88.1 | 85.1 | 89.0 | 86.6 | 87.3 | 83.3 | 86.0 | 82.9 | 83.3 | 78.4 | 83.7 | 85.6 | 85.0 |
| gaussian_medium | 84.1 | 86.4 | 87.3 | 85.7 | 85.8 | 86.0 | 84.8 | 88.1 | 90.4 | 87.9 | 88.2 | 86.1 | 81.3 | 84.0 | 82.0 | 85.9 |
| gaussian_small | 91.5 | 91.2 | 89.7 | 92.0 | 86.9 | 87.6 | 88.1 | 88.0 | 92.1 | 88.8 | 90.5 | 88.7 | 85.8 | 86.1 | 86.4 | 87.1 |

Table 4.3: Performance of GBDT estimators trained on the `mixture` tasks; from the top: $R^2 \uparrow$ and $MSE \downarrow$ (scaled ×100).

| max_depth | **3** | | | | **4** | | | | **6** | | | | **9** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n_trees | **100** | **200** | **400** | **700** | **100** | **200** | **400** | **700** | **100** | **200** | **400** | **700** | **100** | **200** | **400** | **700** |
| mixture_large | 84.9 | 86.3 | 86.8 | 86.9 | 83.6 | 84.5 | 84.7 | 84.8 | 81.3 | 81.5 | 81.5 | 81.5 | 77.3 | 77.4 | 77.4 | 77.4 |
| mixture_medium | 86.4 | 87.8 | 88.1 | 88.2 | 87.5 | 88.2 | 88.5 | 88.5 | 85.6 | 86.1 | 86.1 | 86.1 | 81.6 | 81.7 | 81.7 | 81.7 |
| mixture_small | 89.7 | 91.1 | 90.8 | 90.4 | 90.0 | 90.0 | 89.7 | 89.4 | 87.6 | 87.6 | 87.5 | 87.5 | 85.2 | 85.2 | 85.2 | 85.2 |
| mixture_large | 12.9 | 11.7 | 11.3 | 11.2 | 14.0 | 13.2 | 13.0 | 13.0 | 15.9 | 15.8 | 15.7 | 15.7 | 19.3 | 19.3 | 19.3 | 19.3 |
| mixture_medium | 12.5 | 11.2 | 10.9 | 10.8 | 11.5 | 10.9 | 10.6 | 10.6 | 13.2 | 12.8 | 12.8 | 12.8 | 16.9 | 16.8 | 16.8 | 16.8 |
| mixture_small | 10.5 | 9.1 | 9.4 | 9.8 | 10.2 | 10.2 | 10.5 | 10.8 | 12.6 | 12.6 | 12.8 | 12.8 | 15.1 | 15.1 | 15.1 | 15.1 |

Table 4.4: Performance of FNN estimators trained on the `mixture` tasks; from the top: $R^2 \uparrow$ and $MSE \downarrow$ (scaled ×100).

| layer_count | **1** | | | | **2** | | | | **3** | | | | **4** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| neuron_count | **16** | **32** | **64** | **128** | **16** | **32** | **64** | **128** | **16** | **32** | **64** | **128** | **16** | **32** | **64** | **128** |
| mixture_large | 78.5 | 78.3 | 78.7 | 77.3 | 79.3 | 71.7 | 74.7 | 74.2 | 81.5 | 80.7 | 77.1 | 77.4 | 73.7 | 73.9 | 73.7 | 77.7 |
| mixture_medium | 74.2 | 68.5 | 71.7 | 71.6 | 73.9 | 74.1 | 72.6 | 70.7 | 76.9 | 73.8 | 73.4 | 72.3 | 79.6 | 72.1 | 74.5 | 69.0 |
| mixture_small | 64.1 | 63.3 | 61.6 | 64.2 | 66.7 | 80.0 | 64.9 | 65.9 | 73.4 | 75.7 | 64.9 | 77.7 | 81.2 | 76.1 | 75.4 | 76.9 |
| mixture_large | 18.3 | 18.5 | 18.1 | 19.4 | 17.7 | 24.2 | 21.6 | 22.0 | 15.8 | 16.5 | 19.5 | 19.3 | 22.4 | 22.2 | 22.4 | 19.1 |
| mixture_medium | 23.7 | 28.9 | 26.0 | 26.1 | 24.0 | 23.8 | 25.2 | 26.9 | 21.2 | 24.0 | 24.5 | 25.4 | 18.7 | 25.7 | 23.4 | 28.5 |
| mixture_small | 36.6 | 37.5 | 39.2 | 36.5 | 34.0 | 20.4 | 35.8 | 34.9 | 27.2 | 24.8 | 35.8 | 22.7 | 19.2 | 24.4 | 25.1 | 23.6 |

# 5. Discussion

In this chapter, we discuss research details, limitations, and future work.

**On explainability and causability.** Changing the explanation impacts the causability capabilities of anyone aiming to interpret it, which we defined in Chapter 1. If one can craft the reference dataset so that the shift in distribution is undetectable, but explanations differs drastically, there become trust issues in explainable machine learning. To overcome this challenge, we propose several measures providing feedback about a given explanation instance. Potential future work is to validate methods not only on real-world applications, but also real stakeholders examining their models. Notably, Poursabzi-Sangdeh et al. (2021) discusses a concept of *information overload*, which refers to impeding the user causability capabilities when presenting too many information in explaining a given phenomenon. Knowing that, for example, robustly assessing change in data distribution for hundreds of variables becomes impossible, a question arises: Are variable attribution explanations useful in model analysis for high-dimensional predictive tasks? The curse of dimensionality is evident in explainable machine learning, and we should develop new methods for overcoming this issue.

**On the choice of models and explanations.** We focused on two most common families of predictive models suited for tabular data, which in modern times are tree-ensembles and neural networks. Several parameter configurations are considered not to rely only on defaults, which vastly extends our previous work (Baniecki & Biecek, 2022). We see some association between the model's complexity combined with performance versus the explanation manipulation effectiveness, which relates to bias-variance tradeoff. Unfortunately, no general outcome for all the tasks is visible, so further research on guidelines in this regard is needed. Overall, SHAP became one of the most used explanation, both in research and business. We rely on the widely used implementations of TreeSHAP and GradientSHAP coming from well-established Python packages, but note that the obtained results may not translate to other implementations. One of the main limitations of our study is the choice of explanation instances, parameterized by $K = 10$. We try to reduce the bias connected with their arbitrary choice by proposing a deterministic procedure of selecting explanations for each experimental setting. In practice, we consider on semi-global attribution-importance explanations to minimize randomness and maximize computational efficiency. Therefore, the experiments can be biased with respect to this selection.

**On the explanation performance.** As mentioned next to Definition 3.6, various statistical methods can be used to quantify the association between the magnitudes of change in explanation and in data. Examples are: (1) a Pearson correlation coefficient, which we omit using as it does not necessarily distinguish between two explanations having different gradient of magnitude, and (2) a decision tree, which may capture a nonlinear relationship, but is harder to interpret for one explanatory variable. The linear model with a goodness of fit and a linear coefficient seemed appropriate in our case.

**On the normalization of measures.** In aim to compare the experimental results between distinct predictive tasks, we propose to normalize distances used to measure explanations and data. This assumption becomes a natural limitation of the study. In future work, it would be best to validate other normalization procedures. For manipulation effectiveness, we divide $d_1$ by the sum of SHAP importance values, which relates to the visual change in explanation presented by the column plot (see examples in Figures 4.1 & 4.4). However, one can normalize $d_1$ by the standard deviation of predictions calculated on the validation set, or the mean prediction itself. This would be less intuitive when looking at a single explanation instance, but provides more general framework to compare manipulation effectiveness across different explanations in one predictive task. For Wasserstein distance, we divide $d_w$ by the range of variable's values, which also relates to the visual representation (see examples in Figures 4.3 & 4.6). While it is intuitive for normal-like distributions, it becomes vulnerable to outliers in data and skewed distributions in general. Worth considering is to normalize $d_w$ by the standard deviation of a given variable distribution. Note that normalizing raw values of a variable (before and after the manipulation) is not feasible, as it directly interferes with the change in distribution.

**On the manipulation algorithm.** Our benchmark is limited by the default parameters of the manipulation algorithm set for all 1000 experiments as constant. It would be rather infeasible to account for changing them in one work. Also, for a potential future work, we acknowledge a possibility of using the introduced genetic algorithm to benchmark other explanation methods, and of different black-box models.

# 6. Conclusion

We argue that explanations cannot be applied without proper evaluation, especially in critical domains like medicine where the prediction entails decisions of potentially detrimental effects. A single point of explanation's failure is the reference dataset used for its estimation.[1] Thus, we focus on quantifying how change in background distribution affects SHAP values, contrary to related work. In Chapter 3, we describe measures for evaluating models and explanations, as well as introduce a versatile genetic algorithm that can be used for manipulating them. These methods are then applied to benchmark explanations in Chapter 4.

The main conclusion, apart from the fact that explanations can be manipulated and we ought to interpret them in context of data, is that TreeSHAP of GBDTs present themselves as far more robust than GradientSHAP of FNNs. In most scenarios, the magnitude of data change has positive association with the manipulation effectiveness. Both model complexity and size of data have impact on the explanations' vulnerability but it seem to highly depend on the specifics of the predictive task in question. Therefore, we propose stakeholders use a tree-like meta-model to *globally* evaluate explanations for a given task, and use the measure of explanation performance to *locally* pinpoint the correlation and magnitude of association for a given SHAP instance.

**Broader impact.** Explainable machine learning has a high potential to positively impact society. However, explanations ought to be used with caution, which is called to attention by this work. We provide adversarial scenarios from medicine as an example of a high-stakes decision process, but other applications across finance, research, law, or business become reliant on explanatory model analysis to infer knowledge about data. A proper evaluation scheme is required to adopt these in practice responsibly. We believe this research can guide various stakeholders apparent in the machine learning domain to better understand and interpret model explanations.

[1] In computer science, a single point of failure (SPOF) can be defined as a part of a system that, if it fails, will stop the system from working, e.g. a single server hardware, or software functionality.

# Bibliography

Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, 298: 103502, 2021.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity Checks for Saliency Maps. In *Neural Information Processing Systems (NeurIPS)*, 2018.

Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging Tests for Model Explanations. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Ulrich Aivodji, Hiromi Arai, Olivier Fortineau, Sébastien Gambs, Satoshi Hara, and Alain Tapp. Fairwashing: the risk of rationalization. In *International Conference on Machine Learning (ICML)*, 2019.

David Alvarez Melis and Tommi Jaakkola. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Neural Information Processing Systems (NeurIPS)*, 2018.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Christopher Anders, Plamen Pasliev, Ann-Kathrin Dombrowski, Klaus-Robert Müller, and Pan Kessel. Fairwashing explanations with off-manifold detergent. In *International Conference on Machine Learning (ICML)*, 2020.

Daniel W. Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(4):1059–1086, 2020.

Sercan Ö. Arik and Tomas Pfister. TabNet: Attentive Interpretable Tabular Learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra MojsiloviÄ‡, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John T. Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. AI Explainability 360: An Extensible Toolkit for Understanding Data and Machine Learning Models. *Journal of Machine Learning Research*, 21(130):1–6, 2020.

Ulrich Aïvodji, Hiromi Arai, Sébastien Gambs, and Satoshi Hara. Characterizing the risk of fairwashing. In *Neural Information Processing Systems (NeurIPS)*, 2021.

Hubert Baniecki and Przemyslaw Biecek. Manipulating SHAP via Adversarial Data Perturbations (Student Abstract). In *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. To appear in the proceedings.

Hubert Baniecki, Wojciech Kretowicz, Piotr Piatyszek, Jakub Wisniewski, and Przemyslaw Biecek. dalex: Responsible Machine Learning with Interactive Explainability and Fairness in Python. *Journal of Machine Learning Research*, 22(214):1–7, 2021.

Hubert Baniecki, Wojciech Kretowicz, and Przemyslaw Biecek. Fooling Partial Dependence via Data Poisoning. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2022. To appear in the proceedings.

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.

Umang Bhatt, Adrian Weller, and José M. F. Moura. Evaluating and Aggregating Feature-based Model Explanations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

Przemyslaw Biecek. DALEX: Explainers for Complex Predictive Models in R. *Journal of Machine Learning Research*, 19(84):1–5, 2018.

Przemyslaw Biecek and Tomasz Burzykowski. *Explanatory Model Analysis*. Chapman and Hall/CRC, 2021. ISBN 9780367135591.

Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael Gomes Mantovani, Jan N van Rijn, and Joaquin Vanschoren. OpenML Benchmarking Suites. In *Neural Information Processing Systems (NeurIPS Datasets and Benchmarks Track)*, 2021.

Akhilan Boopathy, Sijia Liu, Gaoyuan Zhang, Cynthia Liu, Pin-Yu Chen, Shiyu Chang, and Luca Daniel. Proper Network Interpretability Helps Adversarial Robustness in Classification. In *International Conference on Machine Learning (ICML)*, 2020.

Leo Breiman. Statistical Modeling: The Two Cultures. *Statistical Science*, 16(3):199–231, 2001.

Raja Chatila, Virginia Dignum, Michael Fisher, Fosca Giannotti, Katharina Morik, Stuart Russell, and Karen Yeung. Trustworthy AI. In *Reflections on Artificial Intelligence for Humanity*, pp. 13–39. Springer, 2021.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning (ICML)*, 2018.

Jiefeng Chen, Xi Wu, Vaibhav Rastogi, Yingyu Liang, and Somesh Jha. Robust Attribution Regularization. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

Sam Corbett-Davies and Sharad Goel. The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning. *arXiv preprint arXiv:1808.00023*, 2018.

Botty Dimanov, Umang Bhatt, Mateja Jamnik, and Adrian Weller. You Shouldn't Trust Me: Learning Models Which Conceal Unfairness From Multiple Explanation Methods. In *European Conference on Artificial Intelligence (ECAI)*, 2020.

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 *Neural Information Processing Systems (NeurIPS)*, 2019.

Ann-Kathrin Dombrowski, Christopher Anders, Klaus-Robert Müller, and Pan Kessel. Towards robust explanations for deep neural networks. *Pattern Recognition*, 121:108194, 2022.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository. *University of California, Irvine, School of Information and Computer Sciences*, 2017.

Dominik Maria Endres and Johannes E Schindelin. A new metric for probability distributions. *IEEE Transactions on Information theory*, 49(7):1858–1860, 2003.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

John Fox and Sanford Weisberg. *An R Companion to Applied Regression*. Sage, Third edition, 2019.

Kazuto Fukuchi, Satoshi Hara, and Takanori Maehara. Faking Fairness via Stealthily Biased Sampling. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of Neural Networks Is Fragile. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

Navdeep Gill, Patrick Hall, Kim Montgomery, and Nicholas Schmidt. A Responsible Machine Learning Workflow with Focus on Interpretable Models, Post-hoc Explanation, and Discrimination Testing. *Information*, 11(3):137, 2020.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting Deep Learning Models for Tabular Data. In *Neural Information Processing Systems (NeurIPS)*, 2021.

Brandon M. Greenwell and Bradley C. Boehmke. Variable Importance Plots-An Introduction to the vip Package. *The R Journal*, 12(1):343–366, 2020.

John J Grefenstette. Genetic algorithms and machine learning. In *Conference on Learning Theory (COLT)*, 1993.

Kazuaki Hanawa, Sho Yokoi, Satoshi Hara, and Kentaro Inui. Evaluation of Similarity-based Explanations. In *International Conference on Learning Representations (ICLR)*, 2021.

Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'ıo, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling Neural Network Interpretations via Adversarial Model Manipulation. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Andreas Holzinger, Georg Langs, Helmut Denk, Kurt Zatloukal, and Heimo Müller. Causability and explainability of artificial intelligence in medicine. *WIREs Data Mining and Knowledge Discovery*, 9(4):e1312, 2019.

Andreas Holzinger, Anna Saranti, Christoph Molnar, Przemyslaw Biecek, and Wojciech Samek. Explainable AI Methods - A Brief Overview. In *Workshop on Extending Explainable AI Beyond Deep Models and Classifiers (xxAI ICML)*, 2022.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A Benchmark for Interpretability Methods in Deep Neural Networks. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning (ICML)*, 2015.

Sergio Jesus, Catarina Belem, Vladimir Balayan, Joao Bento, Pedro Saleiro, Pedro Bizarro, and Joao Gama. How can I choose an explainer? An Application-grounded Evaluation of Post-hoc Explanations. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2021.

Yunzhe Jia, Eibe Frank, Bernhard Pfahringer, Albert Bifet, and Nick Lim. Studying and Exploiting the Relationship Between Model Accuracy and Explanation Quality. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2021.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV (version 1.0). *PhysioNet*, 2021.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):1–9, 2016.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Neural Information Processing Systems (NeurIPS)*, 2017.

Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (Un)reliability of Saliency Methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 267–280. Springer, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Janis Klaise, Arnaud Van Looveren, Giovanni Vacanti, and Alexandru Coca. Alibi Explain: Algorithms for Explaining Machine Learning Models. *Journal of Machine Learning Research*, 22(181):1–7, 2021.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020.

William H Kruskal. Ordinal measures of association. *Journal of the American Statistical Association*, 53(284):814–861, 1958.

Aditya Kuppa and Nhien-An Le-Khac. Black Box Attacks on Explainable Artificial Intelligence(XAI) methods in Cyber Security. In *International Joint Conference on Neural Networks (IJCNN)*, 2020.

Emanuele La Malfa, Rhiannon Michelmore, Agnieszka M. Zbrzezny, Nicola Paoletti, and Marta Kwiatkowska. On Guaranteed Optimal Robust Explanations for NLP Models. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Himabindu Lakkaraju and Osbert Bastani. "How Do I Fool You?": Manipulating User Trust via Misleading Black Box Explanations. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2020.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Faithful and Customizable Explanations of Black Box Models. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2019.

Himabindu Lakkaraju, Nino Arsov, and Osbert Bastani. Robust and Stable Black Box Explanations. In *International Conference on Machine Learning (ICML)*, 2020.

Elizaveta Levina and Peter Bickel. The earth mover's distance is the mallows distance: Some insights from statistics. In *IEEE International Conference on Computer Vision (ICCV)*, 2001.

Yi-Shan Lin, Wen-Chuan Lee, and Z. Berkay Celik. What Do You See? Evaluation of Explainable Artificial Intelligence (XAI) Interpretability through Neural Backdoors. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2021.

Ninghao Liu, Mengnan Du, Ruocheng Guo, Huan Liu, and Xia Hu. Adversarial Attacks and Defenses: An Interpretation Perspective. *ACM SIGKDD Explorations Newsletter*, 23(1):86–99, 2021.

Yang Liu, Sujay Khandagale, Colin White, and Willie Neiswanger. Synthetic Benchmarks for Scientific Research in Explainable Machine Learning. In *Neural Information Processing Systems (NeurIPS Datasets Track)*, 2021.

Scott M. Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Neural Information Processing Systems (NeurIPS)*, 2017.

Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67, 2020.

Aniek F. Markus, Jan A. Kors, and Peter R. Rijnbeek. The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics*, 113:103655, 2021.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, 54(115):1–35, 2021a.

Ninareh Mehrabi, Muhammad Naveed, Fred Morstatter, and Aram Galstyan. Exacerbating Algorithmic Bias through Fairness Attacks. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021b.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

Tim Miller, Piers Howe, and Liz Sonenberg. Explainable AI: Beware of Inmates Running the Asylum Or: How I Learnt to Stop Worrying and Love the Social and Behavioural Sciences. In *Workshop on Explainable Artificial Intelligence (IJCAI XAI)*, 2017. arXiv:1712.00547.

Dang Minh, Xiang H. Wang, Fen Y. Li, and Tan Nguyen. Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review*, pp. 1–66, 2021.

Saumitra Mishra, Sanghamitra Dutta, Jason Long, and Daniele Magazzeni. A Survey on the Robustness of Feature Importance and Counterfactual Explanations. In *Workshop on Explainable AI in Finance (ICAIF XAI)*, 2021. arXiv:2111.00358.

Swati Mishra and Jeffrey M. Rzeszotarski. Crowdsourcing and Evaluating Concept-Driven Explanations of Machine Learning Models. *Proceedings of the ACM on Human-Computer Interaction*, 5(139):1–26, 2021.

Christoph Molnar. *Interpretable Machine Learning.* Self published, 2020. ISBN 9798411463330.

Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. iml: An R package for Interpretable Machine Learning. *Journal of Open Source Software*, 3(26):786, 2018.

Vinod Nair and Geoffrey E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *International Conference on Machine Learning (ICML)*, 2010.

Vedant Nanda, Samuel Dooley, Sahil Singla, Soheil Feizi, and John P. Dickerson. Fairness Through Robustness: Investigating Robustness Disparity in Deep Learning. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2021.

Michael Neely, Stefan Schouten, Maurits Bleeker, and Ana Lucic. Order in the Court: Explainable AI Methods Prone to Disagreement. In *Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI (ICML XAI)*, 2021. arXiv:2105.03287.

Frédérik Paradis, David Beauchemin, Mathieu Godbout, Mathieu Alain, Nicolas Garneau, Stefan Otte, Alexis Tremblay, Marc-Antoine Bélanger, and François Laviolette. Poutyne: A Simplified Framework for Deep Learning, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Amandalynne Paullada, Inioluwa Deborah Raji, Emily M. Bender, Emily Denton, and Alex Hanna. Data and its (dis)contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11):100336, 2021.

Albert Bruno Piek and Evgeniy Petrov. On a Weighted Generalization of Kendall's Tau Distance. *Annals of Combinatorics*, 25(1):33–50, 2021.

Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Wortman Vaughan, and Hanna Wallach. Manipulating and Measuring Model Interpretability. In *CHI Conference on Human Factors in Computing Systems (CHI)*, 2021.

Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi. On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47, 2017.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

Laura Rieger and Lars Kai Hansen. A simple defense against adversarial attacks on heatmap explanations. In *Workshop on Human Interpretability in Machine Learning (ICML WHI)*, 2020. arXiv:2007.06381.

Cynthia Rudin. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence*, 1:206–215, 2019.

Barbara Rychalska, Dominika Basaj, Alicja Gosiewska, and Przemysław Biecek. Models in the Wild: On Corruption Robustness of Neural NLP Systems. In *International Conference on Neural Information Processing (ICONIP)*, 2019.

Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700. Springer, 2019.

Lloyd S. Shapley. A Value for n-Person Games. In *Contributions to the Theory of Games II*, pp. 307–318. Princeton University Press, 1953.

Grace S. Shieh. A weighted Kendall's tau statistic. *Statistics & Probability Letters*, 39(1):17–24, 1998.

Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

Sanchit Sinha, Hanjie Chen, Arshdeep Sekhon, Yangfeng Ji, and Yanjun Qi. Perturbing Inputs for Fragile Interpretations in Deep Natural Language Processing. In *Workshop on Analyzing and Interpreting Neural Networks for NLP (EMNLP BlackboxNLP)*, 2021. arXiv:2108.04990.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling LIME and SHAP: Adversarial Attacks on Post Hoc Explanation Methods. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2020.

Dylan Slack, Sophie Hilgard, Himabindu Lakkaraju, and Sameer Singh. Counterfactual Explanations Can Be Manipulated. In *Neural Information Processing Systems (NeurIPS)*, 2021a.

Dylan Slack, Sophie Hilgard, Sameer Singh, and Himabindu Lakkaraju. Reliable Post hoc Explanations: Modeling Uncertainty in Explainability. In *Neural Information Processing Systems (NeurIPS)*, 2021b.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise. In *Workshop on Visualization for Deep Learning (VDL ICML)*, 2017. arXiv:1706.03825.

David Solans, Battista Biggio, and Carlos Castillo. Poisoning Attacks on Algorithmic Fairness. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2020.

Mateusz Staniak and Przemysław Biecek. Explanations of Model Predictions with live and breakDown Packages. *The R Journal*, 10(2):395–409, 2018.

Erik Štrumbelj and Igor Kononenko. An Efficient Explanation of Individual Classifications using Game Theory. *Journal of Machine Learning Research*, 11(1):1–18, 2010.

Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3):647–665, 2014.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In *International Conference on Machine Learning (ICML)*, volume 70, pp. 3319–3328, 2017.

Sam-Zabdiel Sunder-Samuel, Vidhya Kamakshi, Namrata Lodhi, and Narayanan Krishnan. Evaluation of Saliency-based Explainability Method. In *Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI (ICML XAI)*, 2021. arXiv:2106.12773.

Ruixiang Tang, Ninghao Liu, Fan Yang, Na Zou, and Xia Hu. Defense against explanation manipulation. *Frontiers in Big Data*, 5, 2022.

Richard Tomsett, Dan Harborne, Supriyo Chakraborty, Prudhvi Gurram, and Alun Preece. Sanity Checks for Saliency Metrics. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Leonid Nisonovich Vaserstein. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, 5(3):64–72, 1969.

Giulia Vilone and Luca Longo. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76:89–106, 2021.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, 17(3):261–272, 2020.

Zifan Wang, Haofan Wang, Shakul Ramkumar, Piotr Mardziel, Matt Fredrikson, and Anupam Datta. Smoothed Geometry for Robust Attribution. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Walt Woods, Jack Chen, and Christof Teuscher. Adversarial explanations for understanding image classification decisions and improved neural network robustness. *Nature Machine Intelligence*, 1(11):508–516, 2019.

Katarzyna Woznica and Przemyslaw Biecek. Does imputation matter? Benchmark for predictive models. In *Workshop on the Art of Learning with Missing Values (ICML Artemiss)*, 2020. arXiv:2007.02837.

Katarzyna Woźnica, Mateusz Grzyb, Zuzanna Trafas, and Przemysław Biecek. Consolidated learning – a domain-specific model-free optimization strategy with examples for XGBoost and MIMIC-IV. *arXiv preprint arXiv:arXiv:2201.11815*, 2022.

Alden H. Wright. Genetic Algorithms for Real Parameter Optimization. In *Foundations of Genetic Algorithms*, volume 1, pp. 205–218. Elsevier, 1991.

Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (In)fidelity and Sensitivity of Explanations. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, and Ting Wang. Interpretable Deep Learning under Fire. In *USENIX Security Symposium*, 2020.

Xingyu Zhao, Wei Huang, Xiaowei Huang, Valentin Robu, and David Flynn. Baylime: Bayesian local interpretable model-agnostic explanations. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021.

Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. Do Feature Attribution Methods Correctly Attribute Features? In *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. To appear in the proceedings.

## List of abbreviations

- AI: Artificial Intelligence
- XAI: Explainable Artificial Intelligence
- SHAP: SHapley Additive exPlanations
- LIME: Local Interpretable Model-agnostic Explanations
- PVI: Permutation-based Variable Importance
- MUSE: Model Understanding through Subspace Explanations
- ROAR: Remove And Retrain
- MIMIC: Medical Information Mart for Intensive Care
- ICD: International Classification of Diseases
- WHO: World Health Organization
- GBDT: Gradient Boosting Decision Trees
- FNN: Feedforward Neural Network
- ReLU: Rectified Linear Unit
- IG: Integrated Gradient
- AUC ROC: Area Under the Receiver Operating Characteristic Curve
- MSE: Mean Squared Error
- ME: Manipulation Effectiveness
- RME: Ranking Manipulation Effectiveness
- ANOVA: Analysis of Variance

# List of figures

# List of tables