Looking at Explainable AI Methods Through The Lens of Causality

by

Seyed Arad Ashrafi Asli

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

With machine learning models becoming more complicated and more widely applied to solve real-world challenges, there comes the need to explain their reasoning. In parallel with the advancements of deep learning methods, Explainable AI (XAI) algorithms have been proposed to address the issue of transparency and shed some light on the decisions of black box machine learning models. Many works try to categorize and compare XAI methods to one another, but they usually provide a subjective outlook. The first contribution of this research is proposing a quantifiable approach to compare XAI methods based on causal inference.

LIME and SHAP are two of the most popular XAI methods. The result of these two algorithms is a ranking of feature importance. In a sense, they seek to demonstrate how important a feature is in predicting the outcome. We thoroughly question this pipeline of training a black box deep learning model and then explain it afterward using XAI methods. Generating a diverse set of experiments with various causal relationships, we quantify how much the output of LIME and SHAP aligns with the causal relationships at hand. The second contribution of this work is to use our suggested quantifiable framework in action to see how aligned the output of these widely used XAI methods is according to the causal baseline.

A human being is a part of the whole, called by us "Universe," a part limited in time and space. He experiences himself, his thoughts and feelings as something separate from the rest — a kind of optical delusion of his consciousness. This delusion is a kind of prison for us, restricting us to our personal desires and to affection for a few persons nearest to us. Our task must be to free ourselves from this prison by widening our circle of compassion to embrace all living creatures and the whole of nature in its beauty.

– Albert Einstein, 1950.

We made a prison out of our thoughts and feelings, in which we experience ourselves as separate. Dig a hole in this prison and free yourself to see what you really are.

– Rumi, 1200s

Acknowledgements

Thanks to my supervisor, Denilson Barbosa, who patiently stood by my side and reminded me to find love and joy while working on my thesis. Thank you for showing me, with or without words, that if those aspects are missing, there would be no meaningful work.

Thanks to Scotiabank and the Department of Computing Science for funding this project and trusting me. This research flourished with their support.

Thanks to André dos Santos, who ignited my curiosity about this field. I might not have started this research without his encouragement and support in the initial phases of this project.

Thanks to the University of Alberta which accepted me as a part of its unique and valuable family.

Contents

1 Introduction **1**
- 1.1 The need to explain AI models . . . 1
  - 1.1.1 What is XAI . . . 2
  - 1.1.2 What do we mean by explanation? . . . 2
  - 1.1.3 Who needs XAI? . . . 2
  - 1.1.4 How can we get explanations? . . . 3
  - 1.1.5 What is being overlooked in explaining AI models this way . . . 3
- 1.2 Causality . . . 4
  - 1.2.1 Motivation to use Causality . . . 5
  - 1.2.2 Correlation Is Not Causation . . . 7
- 1.3 Research Question . . . 8
  - 1.3.1 A Toy Example . . . 8

2 Background **10**
- 2.1 XAI . . . 10
  - 2.1.1 Model-Based Explanations . . . 11
  - 2.1.2 Example-Based Explanations . . . 11
  - 2.1.3 Attribution-Based Explanations . . . 11
- 2.2 LIME . . . 12
- 2.3 SHAP . . . 14
  - 2.3.1 Additive Feature Attribution (AFA) Methods . . . 14
  - 2.3.2 Shapley Values . . . 15
  - 2.3.3 Desiredness of using Shapley values in XAI . . . 17
  - 2.3.4 Kernel SHAP . . . 17
- 2.4 Causality . . . 18
  - 2.4.1 Graphical Structures . . . 19
  - 2.4.2 Confounder . . . 22

3 Related Work **25**
- 3.1 CXPlain . . . 25
- 3.2 Causal SHAP . . . 25

4 Methodology **27**
- 4.1 Synthetic Data Generation . . . 27
  - 4.1.1 Machine Learning Task . . . 28
  - 4.1.2 Causal Groundtruths . . . 29
- 4.2 Causal Scenarios . . . 29
  - 4.2.1 Causal Scenario One . . . 30
  - 4.2.2 Casual Scenario Two . . . 31
  - 4.2.3 Causal Scenario Three . . . 32
  - 4.2.4 Causal Scenario Four . . . 33
- 4.3 Our Classifiers . . . 34

4.3.1 Single Layer Neural Network . . . 35
4.3.2 Neural Network with Two Hidden Layers . . . 35
4.3.3 Neural Network with Five Hidden Layers . . . 35
4.3.4 A Discussion On The Process Of Choosing These Three Classifiers . . . 36
4.3.5 A Note On Addressing These Three Networks . . . 37
4.4 Explanations . . . 38
4.5 Metrics . . . 38
4.5.1 Absolute Error (AE) . . . 39
4.5.2 Reciprocal Rank (RR) . . . 40
4.5.3 Kendall's Tau . . . 42
4.5.4 A Note On The Chosen Metrics . . . 43

5 Results **44**
5.1 The Impact of Classifiers' Complexity . . . 44
5.2 Scenario One's Results . . . 45
5.3 Scenario Two's Results . . . 49
5.3.1 First Category . . . 49
5.3.2 Second Category . . . 51
5.3.3 Third Category . . . 51
5.4 Scenario Three's Results . . . 52
5.5 Scenario Four's Results . . . 53

6 Conclusion **55**

7 Future Work **58**

References **59**

List of Tables

1.1 Mortality Rate of two different treatments 6
1.2 Mortality Rate of two different treatments based on patients health condition 6

4.1 Single-Layered Neural Network Parameters 35
4.2 Parameters of Neural Network with Two Layers 36
4.3 Parameters of Neural Network with Five Layers 36
4.4 Local Explanations for a Hypothetical Dataset with Two Samples only 38
4.5 Global Explanation for the Hypothetical Explanation 38
4.6 Absolute Error for our four hypothetical XAI results 40
4.7 Example of Reciprocal Ranking for Different Returned Results of One Fixed Query 41
4.8 MRR results for our four hypothetical XAI results 41
4.9 Kendall's Tau result for the hypothetical examples 43

5.1 Accuracy of The Three Chosen Architectures 45
5.2 Results of applying LIME and SHAP to a neural network with two hidden layers that were trained on one of the datasets in scenario one 45
5.3 SHAP and LIME's outcome for a case in the first causal scenario with three features 46
5.4 SHAP and LIME's outcome for a case in the first causal scenario with one relevant feature and four random ones. 47
5.5 SHAP and LIME's outcome for all three levels of NN complexities in a case in the first causal scenario with three relevant features and four random ones. 47
5.6 Coefficients of a logistic regression that is trained on the same dataset as the third example above 48
5.7 Results of SHAP and LIME for a case with eight relevant features and one random feature in scenario one. 48
5.8 SHAP and LIME's explanation for cases in scenario two that treatment has higher causal contribution than all of the confounders 50
5.9 SHAP and LIME's outcome versus causal baseline for a case in the first category of scenario two 50
5.10 SHAP and LIME's outcome versus causal baseline for a case in the second category of scenario two 51
5.11 SHAP and LIME's explanation for cases in scenario two that treatment has a similar causal contribution to confounders . . 51
5.12 SHAP and LIME's outcome versus causal baseline for a case in the third category of scenario two 52

5.13 SHAP and LIME's explanation for cases in scenario two that treatment has smaller causal contribution compared to confounders 52
5.14 Results of applying LIME and SHAP to a neural network with two hidden layers that were trained on the datasets of scenario three 52
5.15 SHAP and LIME's outcome versus causal baseline for a case in the third causal scenario 53
5.16 Results of applying LIME and SHAP to a neural network with two hidden layers that were trained on the datasets in scenario four 54
5.17 SHAP and LIME's outcome versus causal baseline for a scenario in the fourth causal scenario 54

List of Figures

1.1 XAI method learning from the black box 4
1.2 XAI method explaining the black box 5
1.3 Causal Graphs for Two Hypothetical Scenarios with The Same Statistical Information. 7
1.4 A toy example showing how an XAI output can assign feature importance in comparison to a corresponding baseline 9

2.1 Example of An Undirected Graph 19
2.2 Example of A Directed Graph 20
2.3 There is one possible structure for two connected nodes 22
2.4 chain graphical structure 23
2.5 A causal graph showing the confounder structure 23

4.1 Overview of Our Experiments' Causal Architecture 30
4.2 Two causal structures of scenario one causal relations without random nodes 30
4.3 Previous example of scenario one causal relations with random nodes added to it 30
4.4 similar examples of scenario one causal complexity in scenario two of complexity 31
4.5 causal diagram for scenario two of causal complexity with some random nodes 31
4.6 similar number of nodes in the previous scenarios' examples in scenario three of complexity 32
4.7 causal diagrams for scenario three of causal complexity with some random nodes 33
4.8 similar number of nodes in the previous scenarios' example, this time with scenario four of complexity 33
4.9 Examples of causal graph for scenario four of causal complexity with some random nodes 34

5.1 Overview of Our Experiments' Results 44

Chapter 1

Introduction

Deep Neural Networks (DNNs) are becoming prevalent in solving real-world problems and automated decision-making. DNNs are being used in image and language processing, self-driving cars, drug discoveries, personalized medicine, detecting crop disease, and so on ([30], [35], [36]).

DNNs are a family of Machine Learning (ML) systems that use an extensive set of parameters to model and find solutions for complex problems. The current trend in using DNNs encourages higher complexity. This higher complexity enables them to model more complicated patterns from the observed data. So, neural networks that are a combination of many parameters and non-linear functions are preferred, and they are capable of reaching high accuracies ([4], [8], [31]). However, the more complicated the design of the networks is, the more difficult it gets to explain them in a human-understandable way ([4]). That is why they are also referred to as black box models.

1.1 The need to explain AI models

These black-box machine learning systems are often involved in sensitive decision-making scenarios, such as in the medical, criminal justice, and financial domains. With more societies and companies allowing machine learning algorithms to make decisions, more questions are raised regarding the essence of these decisions. It is difficult for humans to trust a judgment without an awareness of the underlying thought process of that system. In addition to that, some problematic decisions made by ML systems created distrust about

these systems.

A famous example of this is a case of machine bias in the criminal field. A machine learning system was responsible for assigning recidivism probability of defendants with criminal records [18]. This system was used in some of the courts of the United States to assist the judges until its bias against minorities like people with black skin color was revealed.

1.1.1 What is XAI

The example of machine bias in the criminal justice field is only one of the many critical situations where an AI model, despite its high accuracy, was reasoning in an undesired and harmful way. These behaviors and the incredible increase in using machine learning systems made it necessary to explain why ML models are making certain decisions. As a result the field of Explainable AI (XAI) emerged to address the need to interpret AI models.

1.1.2 What do we mean by explanation?

In general, an explanation can be thought of as anything that assists humans in comprehending the behavior of a black box model. In other words, explanations should be human-understandable. This explanation can take the format of a text, number, image, graph, or anything else.

1.1.3 Who needs XAI?

XAI aims to generate explanations that help users understand how the black box works and why it behaves in a certain way. Different types of users can benefit from these explanations. A user can be a person who wants to use AI in their decision-making procedure. For instance, a business owner that wants to gain new insights might want to know why AI is suggesting a certain strategy.

On the other hand, the user can be a person who is being impacted by AI decisions. An example of this can be a person who is denied a loan. This person might want to know the reason behind their rejection and what to

improve in the future to be granted a loan.

The other users who can benefit from XAI are data scientists, ML researchers, and engineers. XAI can help them debug and modify their ML models to meet their needs. XAI can also help them choose one model over another based on their explanations or how explainable each of them is ([3]).

1.1.4 How can we get explanations?

Markus et al. classified XAI techniques based on two features; the type and scope of explanation [17]. We use the same taxonomy and divide the XAI methods into three types: Model-Based Explanations, Attribution-Based Explanations, and Example-Based Explanations, and two scopes: global and local (We open each of these types in the next chapter (Section 2.1)).

Since we focus on a group of attribution-based explanations called model-agnostic explainers in this research, we present on a high level how these model-agnostic XAI methods learn from a trained black-box and how they explain the black-box model. Figure 1.1 shows how XAI learns from the black-box model's prediction on a dataset. The trained black box model would teach XAI algorithm how it is making predictions of the rows in the dataset. After XAI algorithm finished learning from the black box model, it will be used to explain black box predictions of each row in the dataset (Figure 1.2).

1.1.5 What is being overlooked in explaining AI models this way

Explaining AI models seems very promising and gives legitimacy to the current trend of increasing black-box complexity to yield higher accuracy. We questioned this whole pipeline of having a black box model to explaining it later. We expect that these explanations, although very promising, might be misleading and incorrect.

To see how close these explanations are to reality, we need a real explanation of the system. This real explanation should capture all the underlying factors that impact an outcome in a system. If we know the real explanation of how a system should work, we can evaluate the pipeline of using tools to

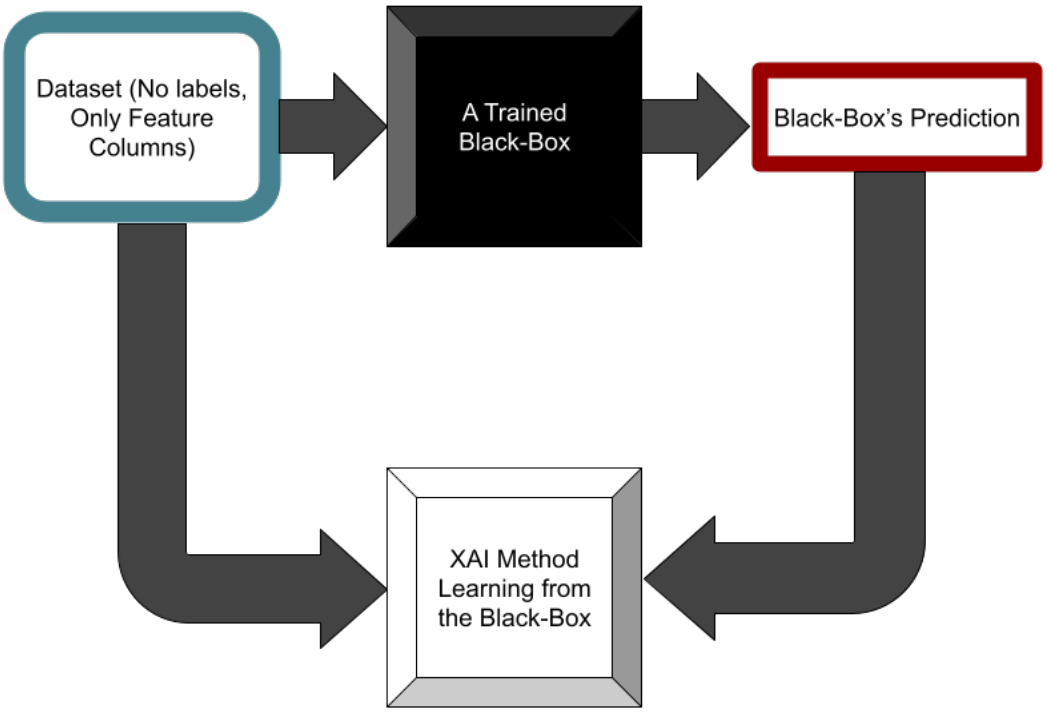


Figure 1.1: XAI method learning from the black box

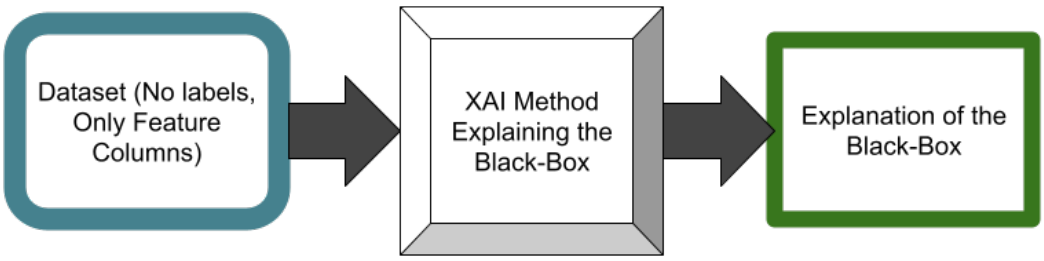


Figure 1.2: XAI method explaining the black box

explain a black box model. Are XAI results close to that real explanation? Are there scenarios in which XAI's results differ from the real explanation? To answer these questions, we needed to visit the realm of causality.

1.2 Causality

Causality provides a set of mathematical tools that might enable us to move from an educated guess to a cause-and-effect relationship between two random variables ([21], [22]). In the context of XAI, causality can provide us with

	Mortality Rate
A	16% (240/1500)
B	19% (105/550)

Table 1.1: Mortality Rate of two different treatments

the reasoning we seek in terms of a real explanation. While both the black box model and some XAI methods function mainly at an associational level, causal inference focuses on finding the causal impact of a random variable on an outcome variable.

1.2.1 Motivation to use Causality

Consider a case where there are two treatments available, and we want to know which treatment is better. We call them treatment A and treatment B, and there are two outcomes surviving or not surviving the disease ([19]). Let's suppose that we have a results like Table 1.1 for this treatments.

Based on this table, it is natural to suppose that treatment A is better than treatment B. However, what if we know people with different health conditions are treated with different treatments (Table 1.2)?

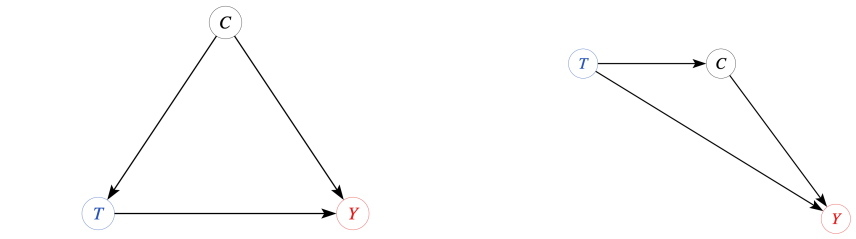
	Mild	Severe	Total
A	15% (210/1500)	30% (30/100)	16% (240/1600)
B	10% (5/50)	20% (100/500)	19% (105/550)

Table 1.2: Mortality Rate of two different treatments based on patients health condition

Each one of the table's columns in Table 1.2 shows the exact opposite of what we infer from Table 1.1. In each category, treatment B is performing better. This phenomenon is called Simpson's paradox. Whenever the marginal probability is different than partial association when controlled for one variable, it is an occurrence of Simpson's paradox. So, can we conclude that

treatment B is better based on the Table 1.2? In fact, it still depends.

Consider the following two scenarios. Scenario one is when treatment B is prescribed for severe cases only because the available resources for this treatment are limited. Scenario two is when taking treatment B takes so long; while waiting for the treatment, many patients' conditions worsen, and their symptoms change from mild to severe. In these two scenarios, the statistical numbers are the same; however, our conclusions can be the opposite. In scenario one, treatment B is the clear winner, and in scenario two, treatment A is preferred because patients won't experience a waiting time that can worsen their symptoms. In causal inference, a certain type of diagram is used, called causal graphs or causal models. These two different scenarios can be shown with two different causal models (Detailed information on Causal graphs can be found on Section 2.4.1). As you see in Figure 1.3, the figure on the left is capturing the first scenario that the condition of a patient (C) is causing the chosen treatment (T). On the other hand, the figure on the right side, is capturing the second scenario in which the chosen treatment is a cause for the condition of a patient. In both scenarios, both treatment and condition are causing the outcome (Y).



[1] [2]

Figure 1.3: Causal Graphs for Two Hypothetical Scenarios with The Same Statistical Information.

1.2.2 Correlation Is Not Causation

Correlation does not imply causation. This is a frequently used statement in the statistics and machine learning literature. In the context of machine

learning, we can even alter it a bit and say prediction does not imply causation! Following the same example we discussed above (Section 1.2.1), if we look at the connection between treatment and outcome without controlling for the common causes, the prediction shows the correlation between treatment and outcome, not the causation. This correlation might rank two treatments in an opposite way of causal relationships. In the same example, it considers the weaker treatment better than the stronger one. ML models that are being deployed in many different fields are able to capture correlations only, and there is no guarantee for causation ([21], [22]). Our prediction might be reasoning in contradiction to a true cause-and-effect relationship. This is a crucial fact that is sometimes overlooked.

1.3 Research Question

In this research, we explored how much the explanation that is provided by popular XAI methods aligned with a real explanation of the system's behavior. We would like to know how trustworthy the pipeline of having a black box model and XAI is. We address this question using popular XAI methods to explain black-box ML classifiers while monitoring the process from a causal viewpoint.

1.3.1 A Toy Example

Before diving deeper into each concept, we present a toy example to introduce our general idea. Assume we have a dataset with three features, W_1, W_2, W_3, and one outcome y. Also, in this case, we have information on the underlying relationship between the features and the outcome. In other words, we know how much the outcome changes if we change any of these features. Let us call this background information the baseline.

This dataset would later be used to train a black box model. After training is finished, XAI methods are applied to this trained black box to generate explanations, as we illustrated in Figure 1.1 and Figure 1.2. Now, we can compare an explanation generated by XAI methods with the baseline and see

how much they agree. Using pie charts in Figure 1.4, we demonstrated a comparison based on a hypothetical case of XAI outcome and a baseline. The numbers for each feature demonstrate how much is the share of that feature in contributing to the outcome.

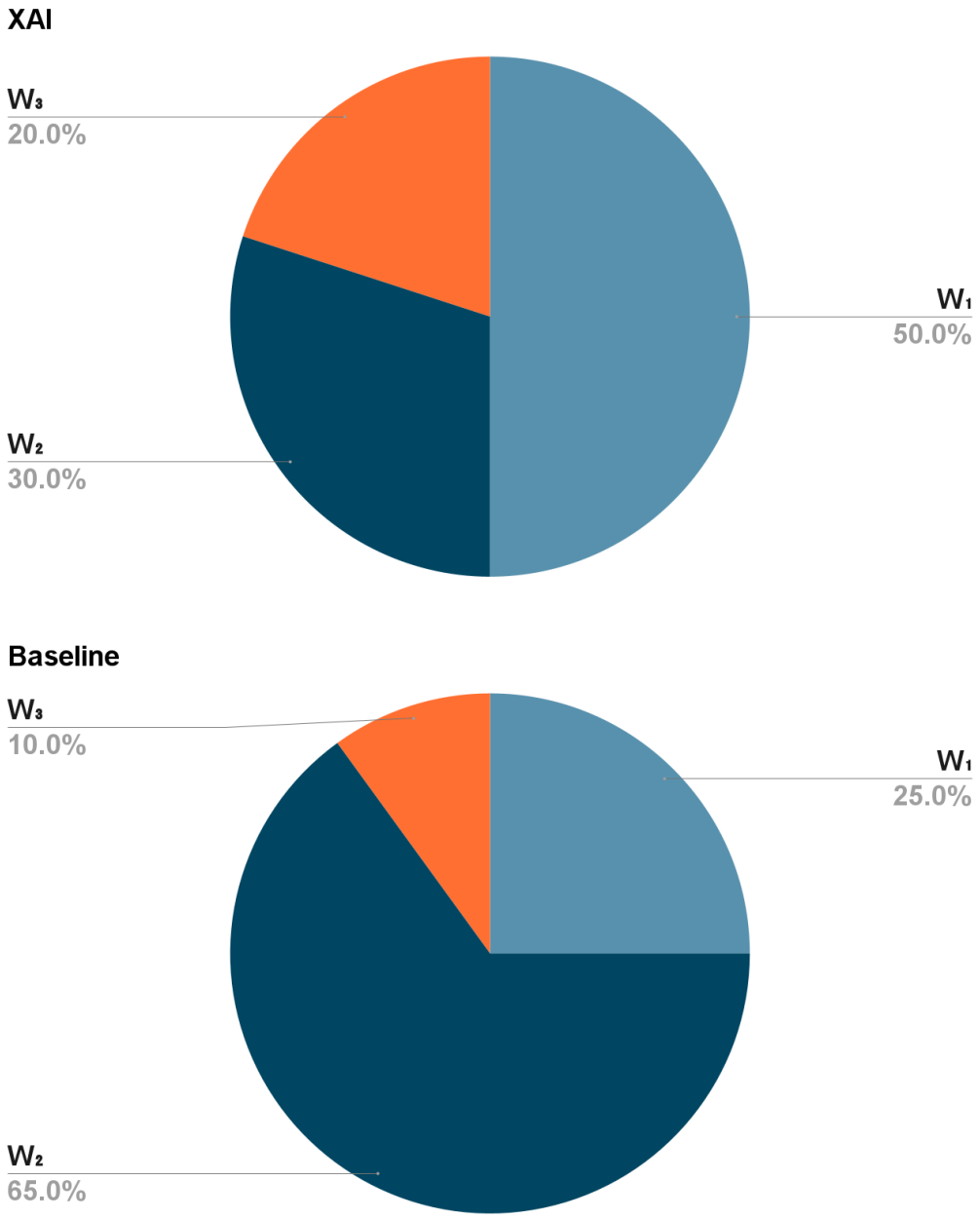


Figure 1.4: A toy example showing how an XAI output can assign feature importance in comparison to a corresponding baseline

Chapter 2

Background

In order to observe how XAI algorithms perform, we need a black-box machine learning model on which we apply XAI methods. Our choice of the black box throughout this project is a neural network (Section 4.3), and our choice of XAI methods are LIME (Section 2.2) and SHAP (Section 2.3). Our main reason for choosing this combination is the high recurrence of this setting in the real-world usage of XAI ([12], [20]). In addition, we use notions of causality in the data generation process to get a baseline of features' relationships to the outcome (Section 2.4).

2.1 XAI

Concepts of explainability and interpretability are used interchangeably in the literature [17]. Some researchers refer to models that are interpretable by design as interpretable models. Decision trees, and regression models, are examples of interpretable models. On the other hand, models that are not interpretable by design (black box models, like deep neural nets) are considered non-interpretable. Explainable AI (XAI) methods are proposed to understand these non-interpretable models. Another important concept is fidelity, which is an abstraction for how faithful is the explanation of the XAI method in its whole domain. Different taxonomies of explanation have been proposed in the literature to categorize XAI methods [3], [10], [32].

Taxonomy proposed by Markus et al. ([17]) summed that all up well. They classified explanations based on the type of the explanation and its scope.

Types of explanation are model-based (Section 2.1.1), attribution-based (Section 2.1.3), and example-based (Section 2.1.2). The scope of the explanation can be local and global. Local explanations explain a specific sample or, in general, a part of the space on which the black box model is being trained. Global explanation explains the behavior of the black box model everywhere ([17]).

2.1.1 Model-Based Explanations

XAI methods that use a model to explain the behavior of the model of interest are in this category. Either the model of interest itself will be used to explain, or a more interpretable model will be utilized. Suppose the model itself is interpretable and is being used to explain. In that case, it will provide a global explanation, which is a special case of having interpretable models by design that we discussed above. If other models are being used to help provide insight and explanation for the model, its scope is usually local ([17]).

2.1.2 Example-Based Explanations

Methods that fall into this category explain the black box's system by using some examples. These examples can be chosen from the dataset or can be newly generated samples (counterfactual examples). The scope of this type of explanation can be local or global. For instance, SP-LIME (Submodular Pick LIME) selects a group of samples from the dataset that it considers essential to explain the behavior of black box globally ([24]).

2.1.3 Attribution-Based Explanations

Without knowing any details about the black box model, this type of explanation assigns a score to each feature based on its contribution to the outcome using the black box's behavior. This process of generating an explanation without the need to know the black box model itself is called model-agnostic explanation.

The scope of this type of explanation is mainly local. Examples are LIME

and SHAP, which provide local insights. Nonetheless, It is possible to make an aggregation of the local explanations into a global explanation ([17]).

We chose SHAP and LIME as the XAI methods in our analysis. This choice is because these two are widely used in the literature and the industry ([1], [7], [16], [33], [34]). We targeted their global explanation by aggregating their local explanations.

2.2 LIME

Local Interpretable Model-Agnostic Explanations (LIME) is a method to provide an explanation for a single prediction of any classifier or regressor. So, LIME tries to explain the behavior of the black-box model around a specific sample of interest, and its objective is to train an interpretable model that behaves similarly to the original model in the vicinity of one instance that we are interested in having explained [24].

To provide an explanation, LIME introduced the idea of interpretable data representation. An important note about these interpretable data representations is that they are not necessarily the features used by the original model. For example, in the field of natural language processing, while an interpretable data representation might be the presence or absence of a word, the feature the model uses could be word embeddings. LIME authors use $x \in \mathbb{R}^{\mathrm{d}}$ notation for the original representation of an instance that is going to be explained and use $x^{'} \in \{0,1\}^{d^{'}}$ for the interpretable binary vector representation.

After defining the interpretable data representation, LIME proposes an idea for the fidelity-interpretability trade-off. This solution addresses the challenge of finding an interpretable model that ensures local fidelity (The meaning of local fidelity is a good approximation to the prediction of the original model locally). LIME defines a class of potentially interpretable models called G. This set of potentially interpretable models can contain linear models or decision trees. A model $g \in G$ has the domain of $\{0,1\}^{d^{'}}$. In other words, the chosen potentially interpretable model uses the presence or absence of interpretable components. We discussed in the previous paragraph that the interpretable

components could be different from the components that the original model uses. LIME uses the word "potentially" interpretable because these models can become too complicated and uninterpretable. For example, a deep decision tree with many branches might become very difficult to interpret. The following objective function has been introduced to guarantee that the potential interpretable models stay as interpretable as possible while satisfying local fidelity:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}}(\mathcal{L}(f, g, \pi_x) + \Omega(g)) \tag{2.1}$$

$\mathcal{L}$ is the loss function that measures how much the interpretable model g was incorrect in approximating f in the local vicinity of x defined by π_x. $\Omega(g)$ is a measure of how complex the interpretable model is. $\Omega(g)$ for decision trees might be their depth, while for linear models, it may be the number of non-zero weights. By minimizing such objective function LIME tries to find the best interpretable model which satisfies local fidelity.

So, LIME aims to minimize the mentioned objective function, which combines locality-aware loss and complexity. The LIME algorithm doesn't make any assumptions about the model f; that is why it is called model-agnostic. It treats the model f as a black box, give samples to it, and see its output. The process of giving samples to the model is also another essential part of LIME's methodology. As previously mentioned, the domains of g and f might be different. While f might use complicated feature representation, g converts those complex features to interpretable representations. For example, consider the interpretable data representation of $x^{'}$, which has a domain of $\{0,1\}^{d^{'}}$ and it is converted from the original feature space of x, which has the domain of $\mathbb{R}^{\mathrm{d}}$.

LIME samples new instances around interpretable representation $x^{'}$ by drawing non-zero elements of $x^{'}$ uniformly at random. The number of these new samples is also determined using a uniform distribution. Then, labels are needed for these newly generated samples to train LIME. In order to acquire labels, interpretable representations would be transferred back to the original

feature space $\mathbb{R}^d$, fed to the model f, and get the label. As we see later in the SHAP algorithm(Section 2.3), SHAP uses the notation of h for a function that transforms this feature between the original feature space and the interpretable representation space. Afterward, the interpretable model g would be trained with these newly generated instances and their labels. Similarly to how it used x for the sample, we want to know its explanation, and $x^{'}$ for the interpretable representation of it, $z^{'}$ is used for newly generated samples in the interpretable feature space, and z is for their transformation back to the original feature space. The weighting function π_x, measures the importance of new instances, $z^{'}$, based on their distance to $x^{'}$. This weighting is used in calculating the loss function $\mathcal{L}$. Their choice for loss function is a locally weighted function as below:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z^{'} \in Z} \pi_x(z)(f(z) - g(z^{'}))^2 \tag{2.2}$$

As one can see, $\pi_x(z)$ weights the error of each generated sample based on the distance of the generated samples to the original data.

2.3 SHAP

SHAP unified most of the previously widely used feature importance methods under the umbrella of additive feature attribution methods. Then, SHAP authors provided their own method, which is based on the game theory concept of Shapley Values [15].

2.3.1 Additive Feature Attribution (AFA) Methods

SHAP uses a similar notation to LIME. f is the original prediction model, and g is the explanation model. Simplified input representations have the same format of $x^{'}$, and the original input representation has the notation of x. SHAP added a new notation of h_x compared to LIME (Section 2.2) which is a mapping function between the original and simplified representation space.

$$x = h_x(x^{'}) \tag{2.3}$$

The other similar notation is $z^{'}$, which is a point in the local vicinity of $x^{'}$. SHAP considers the goal of local XAI algorithm as $g(z^{'}) \approx f(h_x(z^{'}))$ where $z^{'} \approx x^{'}$. SHAP authors unify the goal of all of the six families in the additive feature attribution methods as:

$$g(z^{'}) = \phi_0 + \sum_{i=1}^{M} \phi_i z_i^{'} \tag{2.4}$$

Here $z^{'} \in 0, 1^M$, where M is the number of simplified input features. ϕ_is are scalar quantities that show the importance of each simplified feature, so $\phi_i \in \mathbb{R}$. In this paper, authors unified six other popular XAI methods as AFA methods. These XAI methods are LIME (Section 2.2), DeepLIFT ([27]), Layer-Wise relevance propagation ([2]), Shapley regression value ([14]), Shapley sampling values ([29]), and Quantitative Input Influence ([6]).

2.3.2 Shapley Values

The use of Shapley values ([26]) to compute an explanation was not unknown before SHAP. Before going through those methods, let us first talk about the Shapley values. Shapley value is a concept used in game theory to calculate the contribution of several players in a coalition that leads to a gain or loss. The Shapley value for one player in a game can be calculated using an average expected marginal contribution of one player through all possible combinations of players. Using the idea of Shapley Values in the context of additive feature attribution methods would be something similar to this:

Game: Prediction of a single instance

Gain: Prediction of this instance - Average prediction of all instances

Players: Features that collaborate in the Game, which results in a

Gain

Goal: Calculate the contribution of each player in total Gain inside the Game

Three other methods also used the idea of Shapley Values in the framework they provided to explain model prediction. Shapley regression value, Shapley sampling values, and Quantitative Input Influence. Shapley regression values research, as their name suggests, calculates feature importance for linear models. This calculation is based on retraining the model on different subsets of features. F is used for the set of all features, and S is used to denote subsets. The feature of "interest" is denoted by i. To compute the contribution (gain) of i, a model would be trained with i and another model without i. This would be repeated on all possible combinations of i with other features. If we look at it from the game theory perspective, each of these combinations is a unique cooperation of features as players in the game. The weighted sum of these cooperations makes ϕ_i which is the importance of the feature i in the dataset. ϕ_i would be calculated as:

$$\phi_i = \sum_{S \in F-i} \frac{|S|!(|F| - |S| - 1)}{|F|!} [f_{S \cup i}(x_{S \cup i}) - f_S(x_S)] \tag{2.5}$$

Here x is a sample we want to explain. Set S is the subset of features we consider important in the explanation along with feature i. Another way to say this in the context of game theory is that features in the subset S and feature i are in a coalition. We ignore the contribution of other $F - S - 1$ features in this permutation. $x_{S \cup i}$ keeps only the subset S of all features with the feature of interest i ($S \in F - i$). Similarly, $f_{S \cup i}$ refers to a model that is trained on these filtered data with $S \in F - i$ features along with feature i and ignores the rest of the features. f_S and x_S refers to training a model only with subset S while feature i is withheld. The subtraction of these two models creates a part of the contribution of feature i to the explanation. This needs to be repeated with other subsets of the feature space that can be in cooperation

with feature i (all the possible choices of features from $F - \{i\}$). When we choose a subset S from $F - i$, there are $|S|!$ permutation for this set of assumed important features and $(|F| - |S| - 1)!$ permutation for unimportant features. So, the weight of this set of chosen weights would be $\frac{|S|!(|F|-|S|-1)}{|F|!}$.

In Shapley Regression Values, the mapping function h_x simply means if this feature is included in the model or not (as in the subset S or not). One indicates a feature would be in the model, and zero indicates exclusion from the model. If we consider ϕ_0 as the contribution of an empty set $f_\emptyset(\emptyset)$, then we can see that Shapley Regression Values follow the Additive Feature Attribution methods formula (Equation (2.4)).

2.3.3 Desiredness of using Shapley values in XAI

SHAP defines three desired properties for AFA (Section 2.3.1) XAI methods and proved that only AFA methods that calculate feature importance proportionate to Shapley values satisfy all three of these properties. These properties are local accuracy, missingness, and consistency. Local accuracy requires the explanation model to match the original model's output for the sample of interest. Missingness checks the features that are missing from the original space do not impact the explanation. Consistency states that if a feature contributes more to a model's output among all of the model inputs compared to another, it must have greater explanatory importance in the output of the first one compared to the other one.

Other AFA methods that are not using Shapley values to calculate feature importance satisfy missingness; however, they violate one or both of the other properties. On the other hand, the exact calculation of Shapley values is time-consuming. SHAP proposes the novel idea of kernel SHAP to address this complexity issue.

2.3.4 Kernel SHAP

In Kernel SHAP, LIME algorithm is used to find the Shapley values faster. SHAP proved by setting LIME parameters equal to some specific values, LIME's answer reaches an explanation that is aligned with Shapley values.

LIME's answer under these conditions satisfies all three desired properties and is relatively fast. Thus, this model-agnostic approach became the only AFA method that satisfies all three desired metrics and is not computationally expensive. In Equation (2.1), we demonstrated LIME's formula. The following equation shows the conditions in which LIME's answer would satisfy all three desired properties.

$$\begin{aligned} &\Omega(g) = 0, \\ &\pi_{x'}(z') = \frac{(M-1)}{(Mchoose|z'|)|z'|(M-|z'|)}, \\ &\mathcal{L}(f, g, \pi_x) = \sum_{z' \in Z} [f(h_x^{-1}(z')) - g(z')]^2 \pi_{x'}(z') \end{aligned} \tag{2.6}$$

Setting $\Omega(g) = 0$ means that there would be no penalty on how complex the interpretable model is. $\pi_{x'}$ uses $|z'|$ and M for assigning a score to how close this newly generated instance is to the original data point from the dataset. $|z'|$ is the number of non-zero elements in z' and M is the maximum coalition size based on x'. Similar to LIME, loss function $\mathcal{L}$ is a sum of squared errors, which is weighted by π. From this point onward, whenever we mention SHAP, we are referring to Kernel SHAP.

2.4 Causality

Causality's role in this thesis is in the data generation process, which we will discuss in details later. As we mentioned in the research question (Section 1.3), we want to compare XAI's outcome with a real explanation of the system. Real explanation as we defined in the introduction, captures all the underlying factors that impact an outcome in a system. Generating datasets based on a known causal relationship between features gives us the real explanation that we are looking for. This causal knowledge will serve as a baseline to compare XAI methods. We walk through how we generated datasets with an underlying causal relationship in Chapter 4. In this section, we take a look into causality concepts.

Our logic works in terms of cause and effect. Analyzing, learning and expla-

nation is tied to causal reasoning for us [9]. Despite ML models' outstanding performance, There is no guarantee that these models can learn cause and effect relationships among different features. They are not designed to capture cause-and-effect relationships. As Judea Perl articulates, we do not empower machine learning algorithms with the causal logic tools [22]. Thus, there is no expectation that ML models, from the simplest to the most complex models, can accurately capture existing causal relationships.

In the introduction, with an example, we showed that correlation does not imply causation (Section 1.2). Therefore, by relying solely on correlation, our analysis might be misleading and result in making a wrong decision. Using causality, we want to research how much our decisions while using XAI can be misleading. Let us first define what a causal graph is.

2.4.1 Graphical Structures

Before defining causal graphs, we need to define terminologies such as directed acyclic graphs and Bayesian networks.

Graphs and Related Terminologies

Figure 2.1 is an example of an **undirected graph**. **Nodes** or **vertices** are A, B, C, and D. These nodes are connected together with **undirected edges**. That is why they are called undirected graphs.

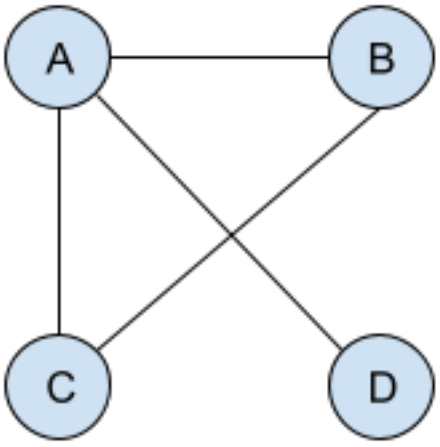


Figure 2.1: Example of An Undirected Graph

On the other hand, if all of the edges are directed, the graph would be considered a **directed graph**. The directed version of the above example can be something like Figure 2.2.

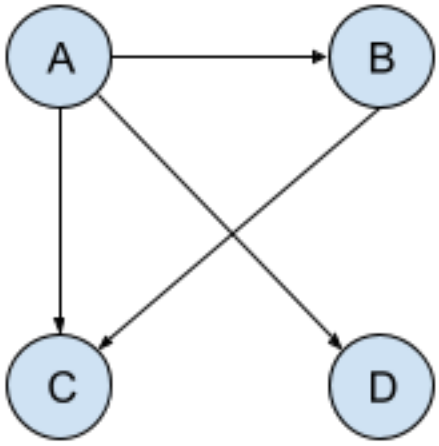


Figure 2.2: Example of A Directed Graph

Adjacent nodes, are nodes that are connected directly with an edge. In this example, A and D are adjacents, A and B are adjacent, but B and D are not. In directed graphs, a **Parent Node** is a node that an edge comes out of, and a **Child Node** is a node that edge comes into it. In the directed graph example, A is the parent of nodes B, C, and D; therefore, B, C, and D are children of A. A **path** exists between two nodes in a graph when there is a sequence of edges between them. This sequence does not need to go exactly in the direction of the edges. For example, there is a path from A to B to C that is aligned with the direction of edges, and there is a path from B to A to D that does not exactly follow the direction of edges. However, there is also a **Directed Path** that needs to go in the direction of edges. So, A to B to C is a directed path while B to A to D is not. Nodes that can be reached from a chosen node via a directed path are **Descendants** of that node, and that node itself is called an **Ancestor**. B, C, and D are all Descendants of A in this graph. If each of these descendants has other children and descendants of their own, those will still be considered descendants of A.

Causal Graph

Directed Acyclic Graph (DAG) structure can be used to show causal relationships. DAG can be a causal graph if its edges are drawn to show a cause-and-effect relationship between the parent node and the child node. In order for a DAG to be a causal graph, besides the causal edges assumption, the local Markov assumption should also hold between independent nodes. The Local Markov assumption states that a node is independent of all of its non-descendants given its parents. In the next paragraph, we express this definition mathematically.

Let $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ be a finite set of random variables in a directed graph $\mathcal{G}$ in a way that descendant nodes have a number higher than their parents. For example, X_4 and X_5 can be children of X_2, but X_2 can not be a child of X_2. Let $\mathbf{V}$ be the set of directed edges (X_i, X_j) in $\mathcal{G}$, with $i, j = 1, 2, \ldots, k-1$ and $j > i$. $j > i$ condition ensures that we will not have any loops and an edge from a node to itself. This graph is called a *directed acyclic graph* (DAG) on $\mathbf{X}$. A *directed path* from X_h to X_k is a sequence $X_h, X_m, \ldots, X_k$ nodes, connected with directed edges (X_i, X_j) in $\mathcal{G}$. The *parents* $Pa(X_j)$ of X_j are those X_i such that $(X_i, X_j) \in \mathbf{V}$. Similarly, the *Children* of X_i are those X_j such that $(X_i, X_j) \in \mathbf{V}$.

If the edges (X_i, X_j) are viewed as X_i is a direct cause of X_j, we say $\mathcal{G}$ is a *causal graph* [21].

Bayesian networks and causal graphs have similar graphical diagrams. Bayesian networks help model joint probabilistic distributions by modeling where there is only a true dependency between two variables. This is similar to a causal graph when they model where there is a causal relationship. Whenever there is a causal relationship, there is a dependency, but the opposite does not hold necessary.

Flow of Association in Graphical Building Blocks

The graphical building blocks that we are going to talk about apply to both causal and non-causal DAGs. We explain them from the perspective of causal-

ity and causal graphs, but a similar explanation applies to bayesian networks in a non-causal graph. With two connected nodes, there is only one possible interconnected network shape. Both graphs in Figure 2.3 are examples of that one possible network.

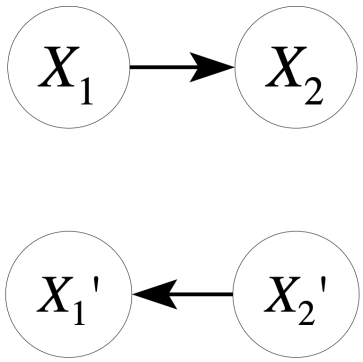


Figure 2.3: There is one possible structure for two connected nodes

In graphs with three nodes or more, there can exist three important structures. Chains, Forks, and Immoralities are the graphical building blocks of causal graphs ([21]). These are considered building blocks because they can determine the dependence or independence of nodes and the flow of dependency or causality in a graph. We will only describe the chain structure since it is the one we used in our research.

Chain

The structure of a chain is shown in Figure 2.4. Here we know that X_1 and X_2 are in a causal relationship. Similarly, X_2 and X_3 are also in a causal relationship. Now the question is, are X_1 and X_3 causally related too? It turns out that they are also causally related because of the flow of causation;

X_1 will cause X_2, and X_2 will cause X_3. Thus, although X_1 is not a direct cause of X_3, a change in X_1 causes a change in X_3 as well.

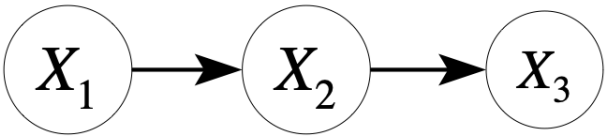


Figure 2.4: chain graphical structure

2.4.2 Confounder

A confounder is a node in a specific causal relationship between three nodes. In Figure 2.5, node C is the confounder. It causes both T (treatment node) and Y (outcome). Treatment node, in general, refers to a feature where we want to know its contribution to an outcome. A confounder is a feature that contributes to the occurrence of both the treatment and outcome nodes.

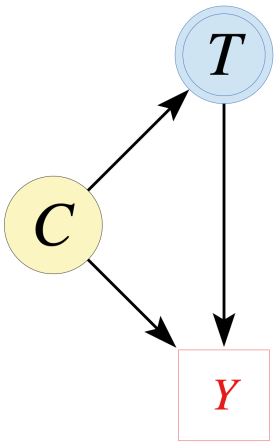


Figure 2.5: A causal graph showing the confounder structure

While estimating the contribution of treatment to the outcome, it is crucial

to take confounders into account. Without taking confounders into account, the estimation of the treatment's contribution to the outcome would be either higher or lower than the original amount. This wrong estimation is due to the fact that the confounder's impact would be mixed with treatment. We will see more of causal graphs and causal structures in the methodology section (Chapter 4). However, before diving into that, we will talk about some related research in the next chapter (Chapter 3).

Chapter 3

Related Work

There are some works in the literature that explored a similar goal to this research. CXPlain ([25]), and Causal SHAP ([11]) are two of the most related ones.

3.1 CXPlain

Using Granger's causality, CXPlain provides a different approach to explain. Granger's causality says there exists a causal relation between a feature X_i and output y if we can better provide a prediction with the presence of feature X_i. In other words, if the absence of random variable X_i decreases the prediction power of output y. From the point of view of Pearl's causality, which we are referring to as causality, Granger's causality is not necessarily referring to a cause-and-effect relationship, and it will still capture correlations. By referring to causality in this research we means Pearl's causality.

CXPlain's authors train a deep neural network using a customized objective based on Granger's causality to explain feature importance. So, they are also from the attribution-based explanation group (Section 2.1.3), which is the same family of approaches as SHAP and LIME.

3.2 Causal SHAP

Using Perl's causality, causal SHAP incorporates concepts of causality into the SHAP (Section 2.3) algorithm. Causal SHAP belongs to the attribution-based

explanations (Section 2.1.3) because the output is similar to SHAP ranking features' importance. While having the same observational dataset, causal SHAP shows how different SHAP's explanations can be when we have different causal graphs (Section 2.4.1). Furthermore, they divide the total effect of each feature into direct and indirect effects of that feature into the outcome.

Causal SHAP also proposes a method for cases where true causal order is unavailable and there is access to only partial causal orders. This is a novel algorithm based on what they call causal chain graphs ([13]). The difference between this research and ours is that they are trying to improve the SHAP algorithm to be more aligned to a causal explanation, while we are analyzing how close is the explanation provided by LIME and SHAP to a causal relationship between the features.

Chapter 4

Methodology

According to our goal to investigate XAI through causality, we generated many synthetic datasets. For generating a synthetic dataset, we first developed a causal relationship between features and the outcome (we also refer to this relationship as the causal baseline for this data). Then, data points were generated based on that causal baseline. Next, we trained neural network classifiers, with different levels of complexity, on those generated datasets. Then, XAI algorithms, SHAP (Section 2.3) and LIME (Section 2.2), were applied to explain the neural network classifier's outcome. After LIME and SHAP returned their feature importance ranking, we compared that ranking with the causal baseline.

4.1 Synthetic Data Generation

We used the DoWhy [1] library to generate our synthetic datasets. DoWhy is one of the most used Python libraries in causal inference. They provided many tools for causality, and we used the help of one of the tools in their library to generate synthetic datasets. SHAP and LIME use a linear function to approximate the black box model's performance around a data point. To make the experiments more favorable and fair for SHAP and LIME, we designed linear causal relationships between the features. Then, we generated datasets based on the defined underlying causal relationships. In other words, the causal relationships in all of the generated datasets are linear. For example,

[1] py-why.github.io/dowhy/

in a case with one treatment and two confounders, our outcome y will have the following linear causal relationship with treatment T and confounders C_0 and C_1:

$$y = c_0T + c_1C_0 + c_2C_1$$

We might have other causal relations in cases with more causal complexities. For example, In the cases with confounders, we will have a separate causal procedure for generating the treatment node. The following expands on the same example with one treatment T and two confounders C_0 and C_1 that we had above. Here, we will have the following generative formula for treatment in addition to the one we have mentioned for the outcome:

$$t = c_3C_0 + c_4C_1$$

In the case of having confounders, this causal generative formula for treatment based on confounders are not of interest to us. We focus only on the generative formula for the outcome since XAI methods are trying to find the explanation for the outcome. However, this is a bit different in our last causal scenario Section 4.2.4. We will discuss this in depth in the related section for each of the causal scenarios. These coefficients, like c_0 to c_4 in the above example, are randomly generated within a range. We used this range to reduce the chances of having large distances between feature causal contributions. This range has a distinct value for each new dataset. In some cases, we explicitly chose these coefficients to test some special and extreme cases.

4.1.1 Machine Learning Task

Binary classification was the chosen task for monitoring SHAP and LIME performance. To have a binary classification dataset, after generating y from the linear causal formula, we converted our numerical outcome to a binary outcome. Then, we took the sigmoid of the number; if it is more than or equal to 0.5, it will be converted to True, and if it is less than 0.5, it will be converted to False. Another way to look at this binarization process is if the numerical outcome is less than zero, we consider it false, and if it is greater or

equal to zero, we consider it true. In this research, we generated data based on four main causal network shapes that we will discuss in more detail in the next section.

4.1.2 Causal Groundtruths

10000 data points were generated for each one of the synthetic datasets based on specified causal relationships. Each of the features inside the dataset is generated with a Gaussian distribution. The mean of the distribution was chosen randomly between -1 and 1. The standard deviation was set to 1.

To add more variety to these causal scenarios, we have added some features that do not have any causal relationship with other features. We call these features random nodes. From the perspective of the causal graph, these random nodes are the nodes that do not have any causal edges with other nodes. The causal relationship between the features will be saved and used as a baseline for XAI's performance.

4.2 Causal Scenarios

We targeted four different scenarios of causal complexity. Figure 4.1 shows the general idea behind them.

Scenario one is the simplest possible causal complexity of features with the outcome because all features have only one causal relation with the outcome. Scenario two, scenario three, and scenario four are designed with a higher level of causal complexity inspired by chain structure (Section 2.4.1) and having a confounder in the graph(Section 2.4.2). In all of these scenarios, we use the word treatment to refer to a specific node in the graph. The treatment node is usually the node in causal analysis that we are interested to know about its contribution to the outcome. In the examples and figures, it is denoted by T. We mentioned the treatment node before in the example of medical treatment's causal effect on mortality with having confounders of the patient's condition (Table 1.1). Now we will dive deeper into each of these four scenarios.

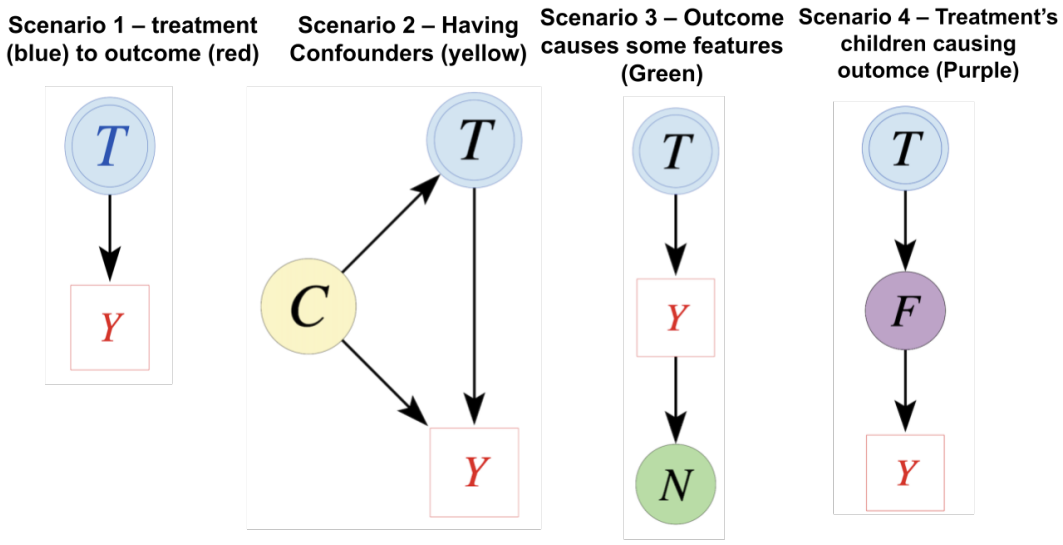


Figure 4.1: Overview of Our Experiments' Causal Architecture

4.2.1 Causal Scenario One

In the simplest form of causal structures, we generated networks with multiple direct causes. In other words, all features have only a direct causal relationship to the outcome and nothing else. Here, the treatment node is one of these direct causes. We added random nodes to see if LIME and SHAP can successfully differentiate between effective features and random ones. Figure 4.2, and Figure 4.3 are examples of this kind of network.

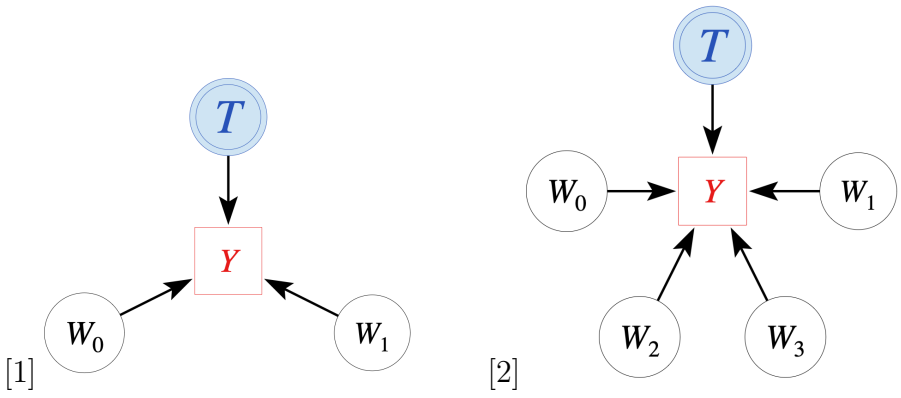


Figure 4.2: Two causal structures of scenario one causal relations without random nodes

4.2.2 Casual Scenario Two

In the second scenario, we generated networks with confounders (Section 2.4.2). We kept the number of treatment nodes fixed at one for simplicity (treatment node, T), and we have only altered the number of confounders and random nodes in different network shapes. So, in this scenario of causal complexity, we have one treatment that has a direct causal effect on the outcome, one outcome, multiple confounders, and multiple non-related nodes. In other words, the only difference in this network shape with the previous scenario is that we have included confounders in the network. Figure 4.4 and Figure 4.5 are examples of the similar number of nodes we had in Figure 4.2 and Figure 4.3, but with scenario two of causal complexity.

Real-world examples of these causal structures are abundant. We can still refer to the medical field, that many genetics-related factors can act as confounders and cause both the treatment of choice by doctors and the mortality rate [5], [23], [28].

4.2.3 Causal Scenario Three

This scenario is inspired by the famous example of smoking, lung cancer, and x-ray results. While x-ray results help doctors to detect if the cancer is present or not, they can not be the cause of cancer, and they are simply an outcome. We were curious to see how much importance would be given to the causes

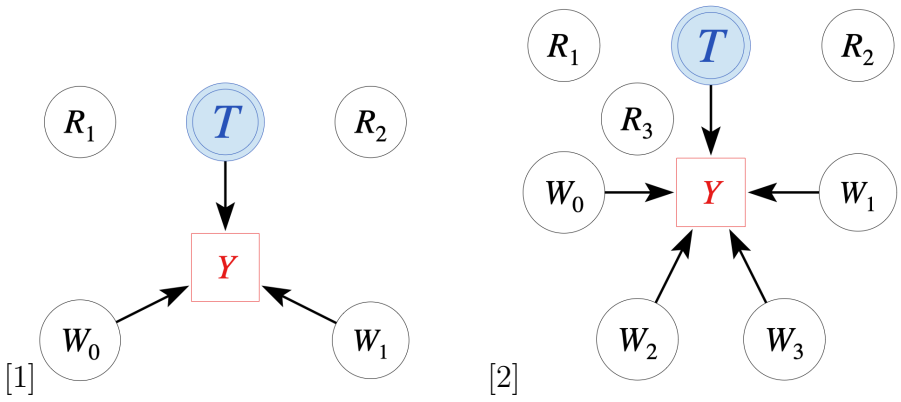


Figure 4.3: Previous example of scenario one causal relations with random nodes added to it

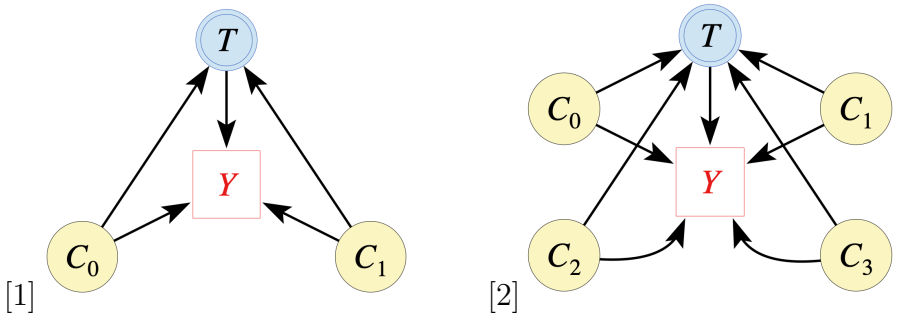


Figure 4.4: similar examples of scenario one causal complexity in scenario two of complexity

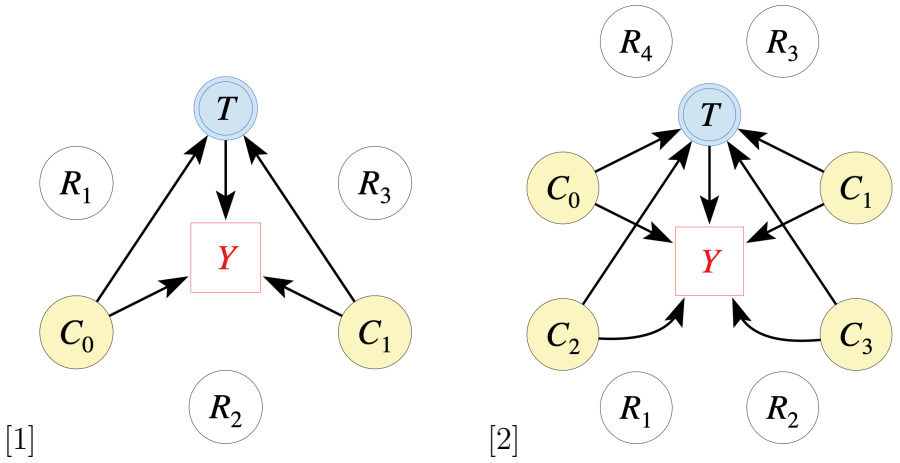


Figure 4.5: causal diagram for scenario two of causal complexity with some random nodes

of the outcome versus the importance of the outcome's children nodes in the XAI methods' explanations. For example, if we truly know all the causes of cancer and have those causes as features in our dataset, how much does the explanation notice them? That was our goal in designing this scenario. To examine what happens to LIME and SHAP's explanation when the outcome is a parent for some feature nodes in the dataset. This scenario, in its most basic form with three nodes, can be seen as a chain structure Section 2.4.1. For simplicity, we kept only one cause for the outcome node, and we increased the children nodes of the outcome. Figure 4.6 and Figure 4.7) depict some examples in this causal scenario with the same number of nodes in the previous scenarios' examples.

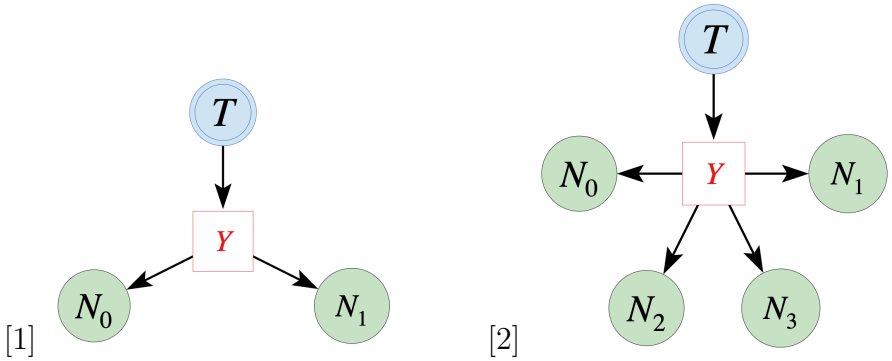


Figure 4.6: similar number of nodes in the previous scenarios' examples in scenario three of complexity

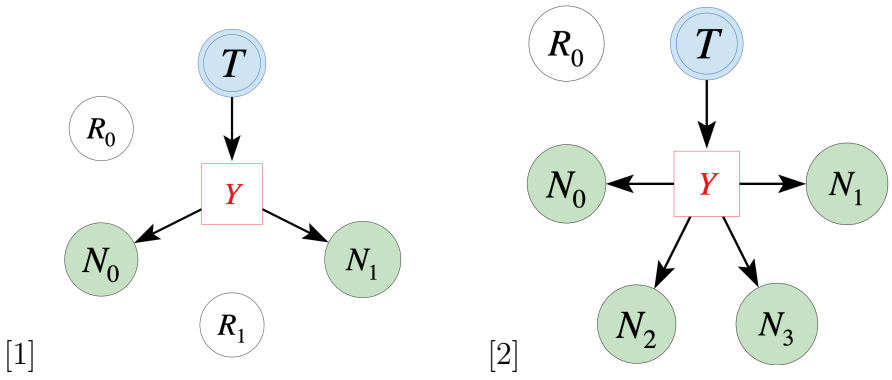


Figure 4.7: causal diagrams for scenario three of causal complexity with some random nodes

4.2.4 Causal Scenario Four

For the fourth scenario, we designed causal relationships in a way to have nodes between the treatment node and the outcome. The treatment node will have children nodes which are the parent nodes for the outcome. We kept the number of treatments fixed at one and changed the number of nodes between treatment and outcome. Similar to the previous scenario, in the most basic form with three nodes, we can see this scenario as a chain. The main difference between this scenario and the previous one is in the positioning of treatment and the outcome in relation to each other. In the figures below (Figure 4.8, Figure 4.9), we show examples with two and four nodes in between treatment and outcome.

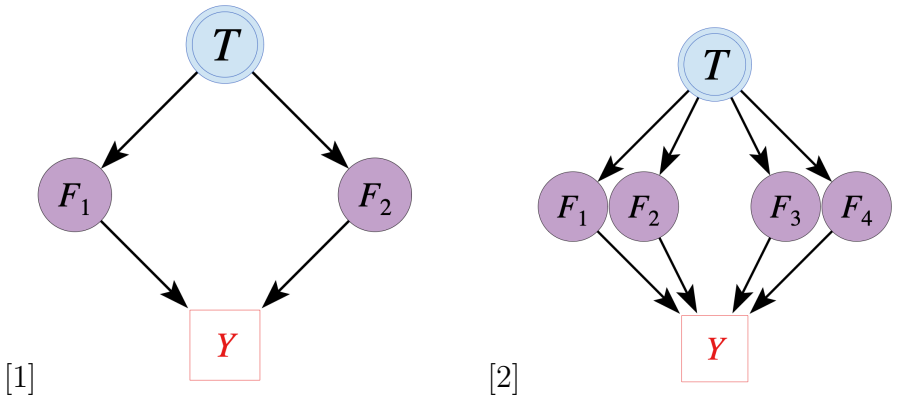


Figure 4.8: similar number of nodes in the previous scenarios' example, this time with scenario four of complexity

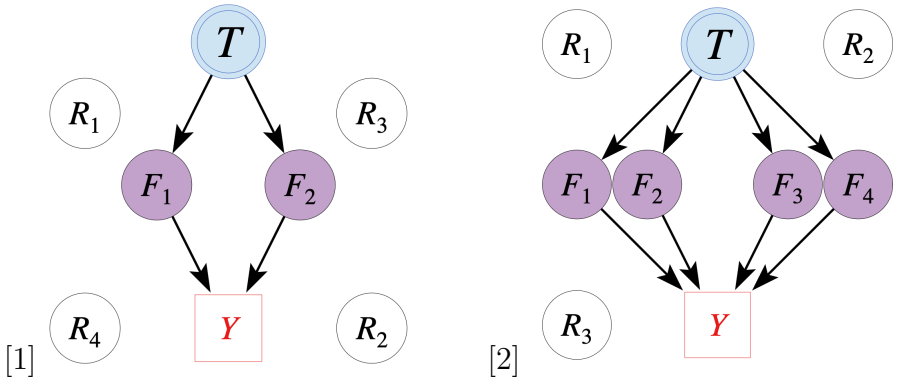


Figure 4.9: Examples of causal graph for scenario four of causal complexity with some random nodes

As mentioned in the synthetic data generation section (Section 4.1), in this scenario, the causal relationship between other features and treatment node matter. This matters because treatment is indirectly causing the outcome, and to find the true causal impact of treatment on the outcome, we use these relationships. For example, consider the left diagram in the Figure 4.8 with two nodes of F_1 and F_2 between the treatment and the outcome. If the generative formulas for those nodes in between are as below:

$$\begin{aligned} F_1 &= c_0 * T \\ F_2 &= c_1 * T \end{aligned} \tag{4.1}$$

And if we have the following generative formula for the outcome:

$$Y = c_2 * F_1 + c_3 * F_2 \tag{4.2}$$

Then, the causal contribution of T on Y would be: $c_0 * c_2 + c_1 * c_3$.

4.3 Our Classifiers

For our black box classifier, we chose three Neural Network (NN) architectures, since they are the most widely used ML architecture in recent years. Inspired by human brains, NNs are mathematical models designed to recognize patterns, predict, and make decisions. We ran our experiments on one single-layer and two multi-layer networks. We use MLPClassifier [2] class from Scikit-Learn library for our Neural Network code. Since generated datasets do not have many features and high-dimensional complexity, we did not find the need to use more advanced NN libraries. To train each of these networks, we kept 80 percent of the data for the training phase and 20 percent for the testing phase.

4.3.1 Single Layer Neural Network

This is our simplest classifier, and it has the least amount of parameters to train among the three classifiers. The table below (Table 4.1) are the parameters of this network that we explicitly set while using the library (the rest of the parameters are kept as default).

[2] https://scikit-learn.org/stable/modules/generated/sklearn.neural$_n etwork.MLPClassifier.html$

Model Parameter	Amount
Hidden Layer Size	100
Batch Size	10
Initial Learning Rate	0.001
Optimizer	Adam
Beta 1	0.9
Beta 2	0.999
L2 Regularization Penalty	0.00001

Table 4.1: Single-Layered Neural Network Parameters

4.3.2 Neural Network with Two Hidden Layers

With two layers, this was our default case of experimenting. Since it was able to capture higher-level complexity than the single-layered NN (Section 4.3.1), and it was faster and easier to train compared to our most complex structure (Section 4.3.3). The only difference between this classifier and the previous one is the hidden layer size. The rest of the parameters are kept intact for the sake of comparability.

Model Parameter	Amount
Hidden Layer Size	(100, 100)
Batch Size	10
Initial Learning Rate	0.001
Optimizer	Adam
Beta 1	0.9
Beta 2	0.999
L2 Regularization Penalty	0.00001

Table 4.2: Parameters of Neural Network with Two Layers

4.3.3 Neural Network with Five Hidden Layers

To have a higher level of complexity to compare the results, we chose a NN with five hidden layers. The rest of the parameters are the same as previous networks (Section 4.3.1, Section 4.3.2).

Model Parameter	Amount
Hidden Layer Size	(100, 100, 100, 100, 100)
Batch Size	10
Initial Learning Rate	0.001
Optimizer	Adam
Beta 1	0.9
Beta 2	0.999
L2 Regularization Penalty	0.00001

Table 4.3: Parameters of Neural Network with Five Layers

4.3.4 A Discussion On The Process Of Choosing These Three Classifiers

The goal of this research is not to find the best parameters that can lead to the most accurate classifier. Even if we have a hypothetical classifier that has an accuracy of hundred percent and is also empowered with all of the causal knowledge, LIME and SHAP will not be able to learn the underlying causal rules. This is because LIME and SHAP are not able to learn causal reasoning by their design (Section 2.2, Section 2.3). The role of the classifier is to label newly generated points in the local vicinity of the point of interest for them (as depicted in Figure 1.1). Let's say that we have two perfect classifiers in terms of accuracy. One of them functions based on causal knowledge, while the other functions based on correlations. Because the labels these two classifiers will generate for LIME and SHAP are the same, there is no way that SHAP and LIME learn anything more from the first classifier, which is empowered by causal knowledge. LIME and SHAP will find the simplest possible way to explain how those limited numbers of points are labeled in that locality. Using correlations, they fit an interpretable model like regression family to the generated points around the point of interest. In the same way that we can not expect regression models or decision trees to learn causal relationships([22]), we can not expect LIME and SHAP to do so.

As mentioned in the introduction of our research question (Section 1.3), our goal is to see how aligned the outcome of LIME and SHAP is compared to the causal baseline and if this explanation can lead to a misleading analysis. So,

we chose NNs and their parameters in a way that leads to very high accuracy on the generated dataset. We want to overfit networks to our data to have a better teacher for XAI methods to explain these specific datasets, and the generalizability of the NN is not important to us. Because of the small size of the feature space in our generated datasets, which is usually less than thirty, and the linear relation between features and the outcome, even a simple logistic regression can yield high accuracy. To have a classifier with higher accuracy, we chose these three NNs and tuned their parameters until we had a network with almost perfect accuracy (the most complex architecture). In this way, we could also explore how much the accuracy of classifiers can impact LIME and SHAP's outcome while one of them perfectly fits the data. After finding this setting that yields perfect accuracy in most datasets, we did not need to explore more architectures and parameters because we had the perfect teacher we were looking for and other reasons mentioned above. The accuracy of these three classifiers will be reported in the next chapter Section 5.1.

4.3.5 A Note On Addressing These Three Networks

We will refer to the network with one layer as our simple network, the network with two layers as our medium network, and the network with five layers as our complex network. We do this to simplify addressing these three architectures later on. By using these terminologies, we mean in relation to each other. When we say complex, we do not intend to say this is a complicated network compared to other NNs that researchers use nowadays. Even our complex architecture is very simple compared to state-of-the-art DNNs. However, because of our dataset's low dimension, all three networks have more than enough complexity to learn patterns and reach high accuracies.

4.4 Explanations

As we saw in the background section (Chapter 2), SHAP and LIME provide explanations locally. In order to have a global explanation of the black box algorithm, SHAP averages over all local explanations. In averaging, it takes

the absolute value of local feature importance. As an example of this process, we demonstrated it for a hypothetical dataset with only two samples that contains three features each (A, B, C):

	A	B	C
data 1	0.3	-0.5	0.8
data 2	-0.7	-0.1	-0.3

Table 4.4: Local Explanations for a Hypothetical Dataset with Two Samples only

In this case, the global explanation would be:

	A	B	C
Global Explanation	$\frac{\|0.3\|+\|-0.7\|}{2}$	$\frac{\|-0.5\|+\|-0.1\|}{2}$	$\frac{\|0.8\|+\|-0.3\|}{2}$

Table 4.5: Global Explanation for the Hypothetical Explanation

We applied the same averaging mechanism to LIME's explanations to have a global explanation.

4.5 Metrics

We used three metrics, and we believe that the combination of these three can give us an informative analysis. Before describing our metrics in detail, we briefly link some of the information in the other sections of the chapter since we are going to use them to unpack the details of our metrics.

We explained how we generate causal baselines and datasets based on causal relationships (Section 4.1.2). The causal baseline for each dataset is sorted based on the features' importance. This causal baseline can be shown using a vector. For example, consider a case that has two random features and four features that contribute to the outcome (six in total). A causal baseline can be something like:

$$[w_1\text{: } 0.5,\ w_2\text{: } 0.35,\ w_3\text{: } 0.1,\ w_4\text{: } 0.05,\ rand_1 : 0,\ rand_2\text{: } 0]$$

The XAI explanation for each dataset can also be shown using vectors as well. To demonstrate how metrics work and their strengths and weaknesses, we will use four different hypothetical XAI results.
XAI output 1: [w_1: 0.3, w_2: 0.2, $rand_1$: 0.2, w_3: 0.1, w_4: 0.1, $rand_2$: 0.1]
XAI output 2: [w_1: 0.5, w_4: 0.15, $rand_1$: 0.15, $rand_2$: 0.1, w_3 : 0.08, w_2: 0.02]
XAI output 3: [w_2: 0.4, w_1: 0.38, w_3: 0.12, w_4: 0.1, $rand_1$: 0, $rand_2$: 0]
XAI output 4: [w_4: 0.5, w_1: 0.2, w_3: 0.2, w_4: 0.1, $rand_1$: 0, $rand_2$: 0]

There can be different ways to interpret the results. The following metrics can capture some of the important ways of interpreting them. Using each of the metrics, we will rank them based on the causal baseline.

4.5.1 Absolute Error (AE)

Since both causal baseline and XAI results can be seen as vectors, any error function that applies to vectors can be used here. We chose to use absolute error for each case in causal scenarios. For each causal scenario, we will use Absolute Error (AE) twice. First, to calculate the error of the target variable (which is denoted by T in the causal graphs). Second, to calculate the total average error, using all features along with T. We call the first one target absolute error and use the notation of $\text{AE}_{\text{target}}$ for it, and the second one is the total absolute error and we use the notation of AE for it. We use the word causal to demonstrate the causal vector in the equation. Similarly, XAI represents the XAI vector. The mathematical notation for target absolute error would be:

$$AE_{target} = |causal[t] - XAI[t]| \tag{4.3}$$

And mathematical notation for total absolute error is:

$$AE = \sum_{i=1}^{n} |causal[i] - XAI[i]| \tag{4.4}$$

The average of these two errors among all the cases in a scenario gives us the mean total absolute error, noted by MAE, and mean target absolute error, noted by $\text{MAE}_{\text{target}}$. We use these notations in the next chapter to show the results.

Table (Table 4.6) shows examples of calculating the total absolute error based on the four hypothetical XAI results.

Results	AE
w_1: 0.5, w_2: 0.35, w_3: 0.1, w_4: 0.05, $rand_1$: 0, $rand_2$: 0	0
w_1: 0.3, w_2: 0.2, $rand_1$: 0.2, w_3: 0.1, w_4: 0.1, $rand_2$: 0.1	0.7
w_1: 0.5, w_4: 0.15, $rand_1$: 0.15, $rand_2$: 0.1, w_3 : 0.08, w_2: 0.02	0.7
w_2: 0.4, w_1: 0.38, w_3: 0.12, w_4: 0.1, $rand_1$: 0, $rand_2$: 0	0.2
w_4: 0.5, w_1: 0.2, w_3: 0.2, w_4: 0.1, $rand_1$: 0, $rand_2$: 0	0.3

Table 4.6: Absolute Error for our four hypothetical XAI results

Averaging all absolute errors of different cases inside each causal scenario gives us the Mean Absolute Error (MAE). Therefore, we will have a target MAE and a total MAE for each causal scenario. Target AE and Target MAE have a range of [0, 1], while Total AE and Total MAE have a range of [0, 2]. The lower the error, the better.

4.5.2 Reciprocal Rank (RR)

Reciprocal rank is a popular method in information retrieval. It is a statistical method to show how good the returned results of a query are. However, it considers the relevance of documents in a binary manner. For each query, only one document is relevant, and it is used to score the result. The positioning of that relevant document among results (its rank) is used to score the returned results. $\frac{1}{Rank}$ is the Reciprocal rank assigned to returned results. The mean Reciprocal Rank (MRR) of a causal scenario is an average over all cases in that scenario. MRR score has the range of (0, 1], with 1 being the best score. The following table (Table 4.7) contains examples of the RR score of one fixed query and different returned results:

Now, the question is how to apply MRR as a metric to our case. We chose MRR to see how successful LIME and SHAP are in identifying the most significant cause of the outcome (the feature with the highest causal contribution). In the case of our example (causal baseline:w_1: 0.5, w_2: 0.35, w_3: 0.1, w_4: 0.05, $rand_1$: 0, $rand_2$: 0) the most significant cause is w_1. So,

Query	Results	RR
d1	d2, d1, d3	$\frac{1}{2}$
d1	d3, d1, d3	$\frac{1}{2}$
d1	d1, d2, d3	1
d1	d1, d3, d2	1
d1	d3, d2, d1	$\frac{1}{3}$
d1	d2, d3, d1	$\frac{1}{3}$

Table 4.7: Example of Reciprocal Ranking for Different Returned Results of One Fixed Query

the ranking process will become something like the table of MRR examples (Table 4.7). In the Table 4.8, we follow the four hypothetical XAI results and rank them based on the causal baseline.

Query	Results	MRR
w_1	w_1, w_2, $rand_1$, w_3, w_4, $rand_2$	1
w_1	w_1, w_4, $rand_1$, $rand_2$, w_3, w_2	1
w_1	w_2, w_1, w_3, w_4, $rand_1$, $rand_2$	$\frac{1}{2}$
w_1	w_4, w_1, w_3, w_4, $rand_1$, $rand_2$	$\frac{1}{2}$

Table 4.8: MRR results for our four hypothetical XAI results

In some real-world scenarios, the most significant feature is interesting for us. For example, financial companies might want to know the most influential feature contributing to the acceptance or denial of a loan application. MRR can show us how successful the explanations of LIME and SHAP are while monitoring the most significant feature.

4.5.3 Kendall's Tau

Kendall rank correlation coefficient or Kendall's Tau coefficient is another statistical method that has the objective of ranking how relevant two documents are. This is one of the most popular methods to calculate the correlation between the ordinal association of two sets. Instead of having a binary approach and looking at the most relevant document, It considers all elements in the order. So, Kendall's Tau can give us a sense of how aligned the order of results

is compared to an ideal ranking of them, while reciprocal rank (Section 4.5.2) only tells how good the rank of the most relevant document is in the result.

Kendall's Tau uses the concepts of concordant and discordant pairs to calculate the correlation of two sets. In our case, one set is the causal baseline which is the ideal ranking, and the other one is XAI results. Let us show the ranking of the first element of a pair in the first ranking with R_1, the ranking of the second element in the first set with $R_1^{'}$. R_2 and $R_2^{'}$ notate the similar concept in the second set. Ranking of a pair of elements in two sets of results is considered concordant if: $R_1 - R_1^{'}$ has the same sign as $R_2 - R_2^{'}$. For example, if feature A is ranked higher than feature B in the first set of results and in the second set, A is also ranked higher than B, they would be considered concordant. If this condition is not met, the pair will be considered discordant. If we represent all of the concordant pairs with variable C and all of the discordant pairs with variable D, Kendall's Tau would be:

$$\tau = \frac{C - D}{C + D} \tag{4.5}$$

Kendall rank correlation coefficient would be a number between -1 and 1. Higher Kendall's Tau coefficient means more similarity between sets. In our cases, random features all have the same importance of zero, and their order relative to themselves does not matter to us. The important part for us is how these are ordered compared to relevant features. So, before calculating the results, we renamed all of these random nodes to the unique name of "rand". Using Kendall's Tau as a metric will give us the following scores (Table 4.9) for our four examples in correlation with baseline (random nodes are all renamed to "rand"):

Results	Kendall's Tau
w_1, w_2, $rand$, w_3, w_4, $rand$	0.0714
w_1, w_4, $rand$, $rand$, w_3, w_2	-0.5
w_2, w_1, w_3, w_4, $rand$, $rand$	0.8572
w_4, w_1, w_3, w_4, $rand$, $rand$	0.6671

Table 4.9: Kendall's Tau result for the hypothetical examples

4.5.4 A Note On The Chosen Metrics

These metrics are chosen from popular and well-known metrics in machine learning and information retrieval fields. It is possible to use other similar metrics that can compare the order of two different rankings or how close two vectors are together. For example, cosine similarity can be used instead of absolute error. Or Spearman's Rho can be used instead of Kendall's Tau. Although using different metrics would yield different results, there should not be inconsistencies in how we analyze the results inside each scenario by averaging many rankings. For example, if the order of XAI's outcome is very different from the baseline, this would be reflected by using Spearman's Rho and Kendall's Tau.

Chapter 5

Results

As discussed in the previous chapter, four common scenarios of causal relationships were studied. More than five hundred datasets were generated based on those causal graphs, and different complexities of neural networks were trained on them. Finally, we applied LIME (Section 2.2) and SHAP (Section 2.3) to them and compared their results with the underlying causal baseline. In this chapter, we represent the results and analysis of those four scenarios. We used Absolute Error (Section 4.5.1, Reciprocal Rank (Section 4.5.2), and Kendall's Tau (Section 4.5.3). These metrics provide a quantifiable measure of how much the explanations of XAI methods are aligned with the causal baseline. On top of that, we break these numbers further and look into the specificity of each scenario. As a brief overview of what has been observed in our experiments, we provide the following figure (Figure 5.1).

Scenario 1	Scenario 2	Scenario 3	Scenario 4
XAI Consistent with Casual Baseline	XAI Inconsistent with Causal Baseline	XAI Inconsistent with Causal Baseline	XAI Inconsistent with Causal Baseline

Figure 5.1: Overview of Our Experiments' Results

5.1 The Impact of Classifiers' Complexity

Before talking about XAI's outcome, we want to report the average accuracy of our chosen classifiers over all the scenarios. Table 5.1 shows the accuracy of the three chosen architectures Section 4.3.

	Accuracy
NN with One Layer	94%
NN with Two Layers	97%
NN with Five Layers	98%

Table 5.1: Accuracy of The Three Chosen Architectures

However, A thought-provoking observation was the irrelevance of the model's complexity on the XAI's results. We will see an example of it in the next section (Table 5.5). This example will represent a general theme that was present in all of the scenarios. Based on our observations, the complexity of the neural network and its higher accuracy are not helping the XAI's output to be closer to the causal baseline. We found the result of all three architectures almost the same. Thus, to keep this chapter more concise, we do not provide examples of all three architectures. The provided results are XAI's outcome when applied to the neural network with medium-level complexity (Section 4.3.2).

5.2 Scenario One's Results

The results of this level of causal complexity were very much aligned with the underlying causal baselines. Some may believe this is not surprising because each of the observed correlations from features to the outcome is also conveying direct causation. However, an important point about this scenario for us was the high level of alignment between XAI methods and the causal baseline. For analyzing the results, it would be good to review that the first two metrics, target MAE and total MAE, are capturing errors. For these two metrics, the lower amount of error represents a better result. However, the other two metrics, Kendall's Tau and MRR, capture scores, and the higher number represents a better result.

XAI Method	MAE	MAE_Target	Kendall's Tau	MRR
LIME	0.311	0.0954	0.859	0.843
SHAP	0.262	0.0814	0.856	0.843

Table 5.2: Results of applying LIME and SHAP to a neural network with two hidden layers that were trained on one of the datasets in scenario one

	T	W_0	W_1
Baseline	0.39	0.40	0.21
LIME	0.31	0.63	0.06
SHAP	0.32	0.63	0.05

Table 5.3: SHAP and LIME's outcome for a case in the first causal scenario with three features

A very high ratio for Kendall's Tau tells that the order of importance was almost always correct. A relatively low absolute error can tell us the magnitude of feature importance given by XAI methods was also very close to the real causal importance of those features. The combination of these two metrics shows how perfect the XAI methods' performance in this scenario was. Also, as you can see, SHAP and LIME are both performing equally well. For example, LIME does slightly better according to Kendall's Tau, which shows it captured the ordering of features slightly better than SHAP, while SHAP performs slightly better according to absolute error. All of these differences in our metrics are small and negligible in our experimental settings. Below are some examples of SHAP and LIME's outcomes and the corresponding causal baseline in this scenario. Both LIME and SHAP were successful in detecting irrelevant features by assigning very small numbers to them. So, in these results, we refrain from including them for simplicity and to make the comparison between relevant features easier.

The first example (Table 5.3) is a case in this scenario with three features, all of which are causally related to the outcome (we have no random nodes). As you can see, both SHAP and LIME have very close predictions, and they are very similar in how they make an error.

For the second example, we chose a case with only one cause but with four

irrelevant features in the dataset. You can see how successful both of them were in recognizing the true feature in play, giving almost all of the importance to that node (Table 5.4).

	T
Baseline	1
LIME	0.95
SHAP	0.97

Table 5.4: SHAP and LIME's outcome for a case in the first causal scenario with one relevant feature and four random ones.

For the third example, we chose a case with three relevant features and four random ones. This was the case that the results of both of these two methods were relatively different from the causal baseline compared to the previous cases. We will only report the results of all three architectures in this case to demonstrate the point we discussed on different complexities of NN architecture (Section 5.1). It is interesting to see that changing the neural network's complexity is not helping the XAI methods at all. In Table 5.5, we show the result of all three levels of the black box complexity, and you can see how similar the results are among complexity levels and between LIME and SHAP.

	T	W_0	W_1
Baseline	0.38	0.38	0.24
LIME_SimpleNN	0.38	0.11	0.46
LIME_MediumNN	0.38	0.11	0.45
LIME_ComplexNN	0.36	0.11	0.48
SHAP_SimpleNN	0.41	0.10	0.48
SHAP_mediumNN	0.41	0.11	0.47
SHAP_ComplexNN	0.41	0.10	0.48

Table 5.5: SHAP and LIME's outcome for all three levels of NN complexities in a case in the first causal scenario with three relevant features and four random ones.

In all of them, we see a similar erroneous pattern. A relatively simple case can give us a misleading analysis in the pipeline of applying XAI methods

on top of a NN. Even more fascinating, below are the coefficients of a simple logistic regression model that is trained on the same dataset. In this case, it is ironic how perfect the outcome of XAI is when it is applied to logistic regression compared to NNs (compare Table 5.6 to Table 5.5).

	T	W_0	W_1
Baseline	0.38	0.38	0.24
Coefficients of a Simple Logistic Regression	0.38	0.38	0.24

Table 5.6: Coefficients of a logistic regression that is trained on the same dataset as the third example above

Applied to this logistic regression model, SHAP and LIME also reflected the exact same weights in their output.

With the last example, we want to illustrate two more findings about this causal scenario. One is about an erroneous behavior, and the other is a strength. The following example illuminates both of the mentioned points altogether (Table 5.7) in a case with nine features that one of which is irrelevant to the outcome, and the eight rest of them are directly causing it. First, let's discuss the error. This error occurs when the causal contribution of features becomes very small and very close to each other. In this case, the output of LIME and SHAP can be slightly different than the causal baseline. The positive point about the results of this causal scenario is that all random features that do not have any causal contribution to the outcome are correctly ranked lower than the relevant features. In this case, we included the score of the random nodes in the table to demonstrate how well LIME and SHAP are scoring this irrelevant feature compared to the other relevant features. As you can see in the table below (Table 5.7), both LIME and SHAP score the irrelevant feature ($Rand_0$) lower than other relevant features. However, to see the erroneous point mentioned in the previous paragraph, some features with a relatively close causal contribution have their ranks slightly differently by LIME and SHAP. For example, you can see how T has a higher score than W_2 in SHAP, while it is the opposite in the causal baseline and LIME.

All in all, explaining black box models that are trained on a dataset with

	T	W_0	W_1	W_2	W_3	W_4	W_5	W_6	$Rand_0$
Baseline	0.216	0.091	0.201	0.022	0.217	0.240	0.007	0.007	-
LIME	0.214	0.087	0.203	0.015	0.215	0.230	0.011	0.017	0.007
SHAP	0.219	0.090	0.195	0.023	0.215	0.237	0.008	0.008	0.003

Table 5.7: Results of SHAP and LIME for a case with eight relevant features and one random feature in scenario one.

this type of causal complexity seems to be consistent with the underlying causal relationships. Remember that the dataset is being generated with a linear generative formula (Section 4.1) to make it more aligned with the linear approximation approach done by LIME and SHAP. Since the explanation is aligned with the underlying causal baseline, we can also conclude that experimentally, the black box model seems to capture relations between features that are aligned with causal relations. So, if you have features in your dataset that, based on your prior knowledge or by consulting with a domain expert, you know are independently causing outcomes, then you can trust the outcome of XAI methods, and you can be more certain that your black box model is behaving according to the true causal relationship. However, we had a surprisingly different set of findings in the other features.

5.3 Scenario Two's Results

In this scenario, we saw an extreme bias towards the treatment node. Due to the confounders' variety of applications and importance in the real world, we have designed many possible experiments. Remember, our chosen architecture has one treatment and multiple confounders (Section 4.2.2). We further divided our experiments in this scenario into three categories to understand this bias better. In the first category, the causal contribution of the treatment node is higher than all of the confounders. Confounders' causal contributions are chosen randomly within a range that has the maximum value of half of the amount of the treatment's causal contribution. In the second category, the causal contribution of confounders is chosen around the treatment's causal contribution. In the last category, we experimented with cases where treatment

is less causally influential than confounders.

5.3.1 First Category

Table 5.8 shows the performance of LIME and SHAP based on our four metrics in the first category.

	MAE	MAE_Target	Kendall's Tau	MRR
LIME	0.60	0.27	0.64	1
SHAP	0.61	0.30	0.90	1

Table 5.8: SHAP and LIME's explanation for cases in scenario two that treatment has higher causal contribution than all of the confounders

SHAP is outperforming LIME based on Kendall's tau metric. However, delving deeper into what happened, it did not seem like an advantage of SHAP over LIME, and both were performing similarly poorly. We observed that LIME and SHAP results are highly biased toward the treatment node. This bias was so high that almost all of the importance was given to the treatment feature, and there was almost no difference between the remaining relevant features, confounders, and irrelevant features. The following example is chosen from a case with one treatment, two confounders, and six random nodes (Table 5.9). This example demonstrates how much the importance of confounders is lost in the SHAP and LIME's explanation. Here feature W_0 has about one-third of the significance of T in the causal baseline, while in the XAI outcome, it has been 50 times smaller than T. Also, irrelevant features, starting with R in the table, have similar scores to confounders. Some unrelated features are scored even higher than relevant confounders. For instance, R_2 and R_3 have higher scores than W_1 in SHAP's output, and R_0 has a higher score than W_1 in LIME's output.

This example clearly shows why MRR has its highest score while MAE is showing a not impressive result. The most significant cause is being overemphasized in XAI's output, so MRR will not capture any error. However, this bias toward one feature and ignorance towards others is captured perfectly with MAE.

	T	C_0	C_1	R_0	R_1	R_2	R_3	R_4	R_5
Baseline	0.788	0.200	0.012	0	0	0	0	0	0
LIME	0.877	0.020	0.016	0.019	0.015	0.014	0.011	0.013	0.015
SHAP	0.960	0.013	0.004	0.004	0.003	0.004	0.004	0.003	0.004

Table 5.9: SHAP and LIME's outcome versus causal baseline for a case in the first category of scenario two

5.3.2 Second Category

In this category, although the causal contribution of treatment and confounders is very close, the treatment node is still dominant in the SHAP and LIME's output. It has a score about ten times higher than the second feature in the ranking. Similar to the first category, irrelevant features have a higher score than relevant confounders. The example below is a case with three confounders and three irrelevant features that will demonstrate those two mentioned points (Table 5.10).

	T	C_0	C_1	C_2	R_0	R_1	R_2
Baseline	0.326	0.240	0.308	0.127	0	0	0
LIME	0.807	0.041	0.047	0.026	0.023	0.033	0.024
SHAP	0.709	0.056	0.060	0.039	0.044	0.050	0.042

Table 5.10: SHAP and LIME's outcome versus causal baseline for a case in the second category of scenario two

Table 5.11 shows a summary of our metrics in this category. We can see a reduction in MRR compared to Table 5.8. The reason is LIME and SHAP are still ranking T as the most important feature, while it is not always aligned with the baseline here.

	MAE	MAE_Target	Kendall's Tau	MRR
LIME	0.78	0.34	0.60	0.81
SHAP	0.77	0.33	0.62	0.80

Table 5.11: SHAP and LIME's explanation for cases in scenario two that treatment has a similar causal contribution to confounders

	T	C_0	C_1	C_2	C_3	C_4	R_0
Baseline	0.063	0.077	0.302	0.072	0.318	0.169	0
LIME	0.390	0.076	0.171	0.042	0.177	0.126	0.020
SHAP	0.280	0.103	0.151	0.092	0.153	0.132	0.089

Table 5.12: SHAP and LIME's outcome versus causal baseline for a case in the third category of scenario two

	MAE	MAE_Target	Kendall's Tau	MRR
LIME	0.81	0.33	-0.12	0.56
SHAP	0.93	0.26	-0.11	0.58

Table 5.13: SHAP and LIME's explanation for cases in scenario two that treatment has smaller causal contribution compared to confounders

5.3.3 Third Category

We reduce the causal contribution of the treatment feature to the point that there are confounders that have five times higher causal contributions in the causal baseline, and the treatment feature has the smallest contribution compared to all other nodes. However, we observed that treatment is still capturing the highest importance in the XAI's output ranking. The example in Table 5.12 makes this point clear. It is from a case with five confounders, one treatment, and one irrelevant feature.

Table 5.13 summarizes the results of this category based on the four metrics. MRR and Kendall's Tau lowered, and they capture the fact that although T is less important than every other feature, it is still being represented as the most influential one.

5.4 Scenario Three's Results

As mentioned in Section 4.2.3, we have only one cause for the outcome. However, our outcome is a cause for multiple nodes. It is surprising how much this high correlation between outcome and its children nodes will remove the focus from the actual cause of the outcome node. The complexity of the neural networks does not seem to help in this scenario either, and all three complexity

levels yield similar results. Table 5.14 shows the medium model's results.

XAI Method	MAE	MAE_Target	Kendall's Tau	MRR
LIME	1.993	0.989	0.095	0.140
SHAP	1.990	0.995	0.075	0.160

Table 5.14: Results of applying LIME and SHAP to a neural network with two hidden layers that were trained on the datasets of scenario three

We are observing the worst possible cases of the target absolute and total absolute errors. Target MAE can be anywhere between 0 and 1, with one being the worst, and here it is almost at its maximum amount. The same happened for the total MAE, which has a maximum error of 2. Also, the low MRR and Kendall's Tau score captured how incorrect the ranking order was. All of these metrics are heavily influenced by the treatment node T. In these scenarios, treatment is the only cause of the outcome, and in the XAI results, node T is being seriously undervalued and covered by the high importance that was given to children nodes of outcome that are not causally contributing to the outcome. Following is an example with seven features. Three of them are random features (denoted by R), three are children of outcome that can not cause outcome (denoted by EC), and we have one true cause (T). See how the true cause is being scored after random nodes.

5.5 Scenario Four's Results

In this scenario, we kept the number of treatments fixed at one and played with the number of nodes between the treatment node and the outcome (Section 4.2.4). Similar to scenario three, the treatment variable was also under-

	T	N_0	N_1	N_2	R_0	R_1	R_2
Baseline	1	0	0	0	0	0	0
LIME	0.012	0.505	0.008	0.366	0.017	0.012	0.009
SHAP	0.007	0.524	0.006	0.397	0.004	0.004	0.004

Table 5.15: SHAP and LIME's outcome versus causal baseline for a case in the third causal scenario

XAI Method	MAE	MAE_Target	Kendall's Tau	MRR
LIME	1.248	0.602	-0.020	0.325
SHAP	1.254	0.603	-0.016	0.334

Table 5.16: Results of applying LIME and SHAP to a neural network with two hidden layers that were trained on the datasets in scenario four

	T	F_0	F_1	F_2	F_3	R_0	R_1	R_2	R_3
Baseline	0.656	0.147	0.021	0.091	0.085	0	0	0	0
LIME	0.012	0.598	0.094	0.178	0.067	0.011	0.012	0.016	0.017
SHAP	0.007	0.638	0.095	0.185	0.058	0.005	0.005	0.005	0.004

Table 5.17: SHAP and LIME's outcome versus causal baseline for a scenario in the fourth causal scenario

valued in the XAI methods' outcome. It seems like a big part of the treatment feature's contribution is distributed amongst the features that are in between the treatment and the outcome. It makes sense for the treatment to have a low score in the presence of all its children nodes. When we have the value of the treatment's children nodes, we can predict the outcome without needing to know the value of the treatment. In other words, although the treatment node causally affects the outcome, it does not have a direct causal contribution. However, another incorrect situation among the results is the ranking of the treatment's children nodes compared to each other. This ranking is also not aligned with their corresponding causal contribution, leading to a very low Kendall's Tau score (Table 5.16).

Here is an example of the XAI methods' outcome when we used a dataset that had four nodes between our treatment and outcome and four irrelevant features. We consider the total causal impact of treatment that is being channeled through its children in the causal baseline instead of its zero direct causal impact (Table 5.17).

Chapter 6

Conclusion

Is it possible that trusting the explanation of black box models misleads us? How much can the model-agnostic explanation of black box models be aligned with a true relationship between features? Are there specific types of scenarios within which we can trust XAI's outcome? Is there a quantifiable way to compare XAI methods?

These are some of the questions we had in mind while starting this research. To address these questions, we needed some scenarios in which we truly know the relationship between the features. Using concepts of causality, we generated such scenarios, and we used this causal knowledge as a baseline for the XAI methods' performance. Four different causal scenarios were generated. Each of these scenarios was inspired by popular causal relationships and real-world examples.

After generating datasets with known underlying relationships, black box models were trained on them. The black box architecture of our choice was the neural network. We trained neural networks with three different complexity levels. After training black box models, we could apply XAI methods to them to compare the results with the causal baseline.

LIME and SHAP are two of the literature's most widely used XAI methods. Thus, we chose them as our XAI methods and applied them to the trained neural networks. In order to compare SHAP and LIME's explanation with the causal baseline, three different metrics were chosen. The chosen metrics were absolute error, reciprocal rank, and Kendall's tau.

Using these metrics, we revealed that we could not trust the explanation of XAI methods that are applied to a black box model. By trust, we mean we can not see it as a real explanation of the features' importance in the system's performance. Explanations were seriously deceiving in some cases. In a general real-world case, part of this problem can be because the black box model, like a neural network, is trained with the assumption of feature independence and is not being empowered by any notions of causality. So, the ML engineer can not say how much of the error is because of the black box model's insufficient accuracy and how much is because of LIME and SHAP. However, in this research, we created a controlled environment to minimize the role of the black box model in the total error. We increased the complexity of the black box to the point that we had perfect accuracy of almost a hundred percent in most cases. So, although we can not say how much of this error is coming from the classifier and how much is coming from the XAI method, we can say that we minimized the error that is coming from the black box classifier.

By having the accuracy of almost one in most cases for our complex classifier, we made the situation as ideal as possible for LIME and SHAP to explain. However, this increase in NN complexity and higher accuracy did not change the XAI methods' results regarding their alignment with a corresponding causal baseline. We saw severe mistakes in the XAI methods' outcome compared to the causal baseline. This error occurred within cases with relatively simple causal scenarios and relatively low numbers of features. The only time that XAI's outcome was aligned with the causal baseline was when all features independently caused the outcome, forming the simplest possible causal scenario. In the cases with higher causal complexity, the XAI's explanation was inconsistent, biased, and in some cases, misleading compared to an actual causal relationship between features.

SHAP and LIME did not show any significant difference in their performance. Although in the SHAP paper, it is claimed that SHAP's explanations are closer to human explanations because SHAP satisfies desired characteristics defined in the paper, it was not the case when we looked into its perfor-

mance compared to LIME from a quantifiable causal viewpoint.

Chapter 7

Future Work

We generated datasets with a linear relationship between their features and the outcome. We mentioned that this makes the environment ideal for LIME and SHAP. Generating datasets with non-linear relationships can provide more results on how LIME and SHAP can be misleading. We leave this for future work due to our time constraints.

Also, it is possible to generate more causal scenarios that we did not cover in this thesis and see how much the explanation provided by LIME and SHAP aligns with the causal baseline in those scenarios. These new scenarios can be designed based on consulting with an ML engineer or a domain expert in a field like finance or medicine. In this way, we can generate more complicated scenarios inspired by real-world cases.

Another interesting future work can be experimenting with smaller and simpler networks that will not overfit the data and see how they can teach LIME and SHAP. Finally, algorithms other than LIME and SHAP can be added to these experiments. For example, CuasalSHAP can be a good candidate to choose and to see how close its explanations are to the causal baseline.

References

[1] A. Adak, B. Pradhan, N. Shukla, and A. Alamri, "Unboxing deep learning model of food delivery service reviews using explainable artificial intelligence (xai) technique," *Foods*, vol. 11, no. 14, p. 2019, 2022.

[2] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, e0130140, 2015.

[3] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020, ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2019.12.012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103.

[4] O.-M. Camburu, "Explaining deep neural networks," *CoRR*, vol. abs/2010.01496, 2020. arXiv: 2010.01496. [Online]. Available: https://arxiv.org/abs/2010.01496.

[5] C. E. Cesta, A. S. Öberg, A. Ibrahimson, I. Yusuf, H. Larsson, C. Almqvist, B. M. D'Onofrio, C. M. Bulik, L. F. de la Cruz, D. Mataix-Cols, *et al.*, "Maternal polycystic ovary syndrome and risk of neuropsychiatric disorders in offspring: Prenatal androgen exposure or genetic confounding?" *Psychological medicine*, vol. 50, no. 4, pp. 616–624, 2020.

[6] A. Datta, S. Sen, and Y. Zick, "Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems," in *2016 IEEE symposium on security and privacy (SP)*, IEEE, 2016, pp. 598–617.

[7] J. Duell, X. Fan, B. Burnett, G. Aarts, and S.-M. Zhou, "A comparison of explanations given by explainable artificial intelligence methods on analysing electronic health records," in *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, 2021, pp. 1–4. DOI: 10.1109/BHI50953.2021.9508618.

[8] R. Eldan and O. Shamir, "The power of depth for feedforward neural networks," in *Conference on learning theory*, PMLR, 2016, pp. 907–940.

[9] P. Gärdenfors, "An epistemic analysis of explanations and causal beliefs," *Topoi*, vol. 9, no. 2, pp. 109–124, 1990.

[10] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, Aug. 2018, ISSN: 0360-0300. DOI: 10.1145/3236009. [Online]. Available: https://doi.org/10.1145/3236009.

[11] T. Heskes, E. Sijben, I. G. Bucur, and T. Claassen, "Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models," *Advances in neural information processing systems*, vol. 33, pp. 4778–4789, 2020.

[12] S. Kawakura, M. Hirafuji, S. Ninomiya, and R. Shibasaki, "Analyses of diverse agricultural worker data with explainable artificial intelligence: Xai based on shap, lime, and lightgbm," *European Journal of Agriculture and Food Sciences*, vol. 4, no. 6, pp. 11–19, 2022.

[13] S. L. Lauritzen and T. S. Richardson, "Chain graph models and their causal interpretations," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 64, no. 3, pp. 321–348, 2002.

[14] S. Lipovetsky and M. Conklin, "Analysis of regression in game theory approach," *Applied Stochastic Models in Business and Industry*, vol. 17, no. 4, pp. 319–330, 2001.

[15] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[16] N. Maltbie, N. Niu, M. Van Doren, and R. Johnson, "Xai tools in the public sector: A case study on predicting combined sewer overflows," in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2021, pp. 1032–1044.

[17] A. F. Markus, J. A. Kors, and P. 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*, vol. 113, p. 103 655, 2021.

[18] S. G. Mayson, "Bias in, bias out," *YAle lJ*, vol. 128, p. 2218, 2018.

[19] B. Neal, *Introduction to causal inference from a machine learning perspective*, https://www.bradyneal.com/Introduction_to_Causal_Inference-Dec17_2020-Neal.pdf, Accessed on: 2022.

[20] H. T. T. Nguyen, H. Q. Cao, K. V. T. Nguyen, and N. D. K. Pham, "Evaluation of explainable artificial intelligence: Shap, lime, and cam," in *Proceedings of the FPT AI Conference*, 2021, pp. 1–6.

[21] J. Pearl *et al.*, "Models, reasoning and inference," *Cambridge, UK: CambridgeUniversityPress*, vol. 19, 2000.

[22] J. Pearl and D. Mackenzie, *The book of why: the new science of cause and effect*. Basic books, 2018.

[23] J.-B. Pingault, F. Rijsdijk, T. Schoeler, S. W. Choi, S. Selzam, E. Krapohl, P. F. O'Reilly, and F. Dudbridge, "Genetic sensitivity analysis: Adjusting for genetic confounding in epidemiological associations," *PLoS genetics*, vol. 17, no. 6, e1009590, 2021.

[24] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[25] P. Schwab and W. Karlen, "Cxplain: Causal explanations for model interpretation under uncertainty," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[26] L. Shapley, "Quota solutions op n-person games1," *Edited by Emil Artin and Marston Morse*, p. 343, 1953.

[27] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, PMLR, Jun. 2017, pp. 3145–3153. [Online]. Available: `https://proceedings.mlr.press/v70/shrikumar17a.html`.

[28] G. D. Smith, D. A. Lawlor, R. Harbord, N. Timpson, I. Day, and S. Ebrahim, "Clustered environments and randomized genes: A fundamental distinction between conventional and genetic epidemiology," *PLoS medicine*, vol. 4, no. 12, e352, 2007.

[29] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and information systems*, vol. 41, no. 3, pp. 647–665, 2014.

[30] M. Taddeo and L. Floridi, "How ai can be a force for good," *Science*, vol. 361, no. 6404, pp. 751–752, 2018. DOI: `10.1126/science.aat5991`. eprint: `https://www.science.org/doi/pdf/10.1126/science.aat5991`. [Online]. Available: `https://www.science.org/doi/abs/10.1126/science.aat5991`.

[31] M. Telgarsky, "Benefits of depth in neural networks," in *Conference on learning theory*, PMLR, 2016, pp. 1517–1539.

[32] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Toward medical xai," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 11, pp. 4793–4813, 2020.

[33] I. R. Ward, L. Wang, J. Lu, M. Bennamoun, G. Dwivedi, and F. M. Sanfilippo, "Explainable artificial intelligence for pharmacovigilance: What features are important when predicting adverse outcomes?" *Computer Methods and Programs in Biomedicine*, vol. 212, p. 106 415, 2021.

[34] C. A. Zhang, S. Cho, and M. Vasarhelyi, "Explainable artificial intelligence (xai) in auditing," *International Journal of Accounting Information Systems*, vol. 46, p. 100 572, 2022.

[35] J. Zhou and F. Chen, *Human and Machine Learning*. Springer, 2018.

[36] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, "Evaluating the quality of machine learning explanations: A survey on methods and metrics," *Electronics*, vol. 10, no. 5, p. 593, 2021.