Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Thu Thao Tran

Explainability vs. Interpretability and methods for models' improvement

Fakultät Technik und Informatik
Department Wirtschaftsinformatik

Faculty of Engineering and Computer Science
Department of Business Informatics

Thu Thao Tran

Explainability vs. Interpretability and methods for models' improvement

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Wirtschaftsinformatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer/in: Prof. Dr. Kai von Luck
Zweitprüfer/in: Dr. Susanne Draheim

Abgabedatum: 10.12.2020

Table of content

Abstract

Thu Thao Tran

Title of the paper:

Explainability vs. Interpretability and different methods for models' improvement

Keyword:

Artificial Intelligence, XAI, explainability, interpretability, post-hoc, transparency

Abstract:

Artificial Intelligence is getting more and more involved in our daily lives, it is being used in almost every area nowadays, from medicinal treatment to autonomous driving. Therefore, another important aspect of artificial intelligence has surfaced in order to assist human in understanding why or how a machine concludes a decision. This aspect is called eXplainable Artificial Intelligence (XAI) or Interpretable Artificial Intelligence. However, it is important to distinguish the difference between "explainable" and "interpretable". Until now, these two terms have been used interchangeably in most of the research papers, this paper attempts to analyze the differences between explainable and interpretable artificial intelligence and hopefully provide people a clear picture of how to use these terms correctly in the future. On the other hand, this paper also attempts to analyze different existing methods or techniques which improve the system's explainability or interpretability. Depending on how deep the users would like to know about the systems they are working with, this paper will provide a more thorough overview when these methods are implemented.

Thu Thao Tran

Thema der Bachelorthesis:

Erklärbarkeit vs. Interpretierbarkeit und verschiedene Methode für die Verbesserung der Modelle.

Stichwort:

Künstliche Intelligenz, XAI, Erklärbarkeit, Interpretierbarkeit, Post-hoc, Transparenz

Zusammenfassung:

Künstliche Intelligenz wird immer mehr in unser tägliches Leben einbezogen und heutzutage in fast allen Bereichen eingesetzt, von medizinischer Behandlung zum autonomen Fahren. Aus diesem Grund haben Experten einen weiteren Aspekt der künstlichen Intelligenz entwickelt, um es verständlicher zu machen, warum oder wie eine Maschine eine Entscheidung trifft. Dieser Aspekt wird als eXplainable Artificial Intelligence (XAI) oder Interpretable Artificial Intelligence bezeichnet. Es ist jedoch wichtig, den Unterschied zwischen „erklärbar“ und „interpretierbar“ auseinander zu halten. Die beiden Begriffe wurden bisher in den meisten Forschungsarbeiten als synonym behandelt. In dieser Arbeit wird versucht, die Unterschiede zwischen erklärbarer und interpretierbarer künstlicher Intelligenz zu analysieren und dadurch den Menschen ein klares Bild darzustellen, wie die beiden Begriffe richtig verwendet werden sollten. Andererseits wird hier auch versucht, verschiedene vorhandene Methoden oder Techniken zu untersuchen, die die Erklärbarkeit oder Interpretierbarkeit des Systems verbessern. Es ist davon abhängig, wie viel die Benutzer über das System, mit denen sie arbeiten, erfahren oder verstehen möchten. Dafür bietet diese Arbeit einen genaueren Überblick über die Implementierung dieser Methoden.

I. Introduction

1. Problems

As mentioned above, artificial intelligence is getting more involved in human's daily lives, especially when technologies are developing at an incredible pace. New ideas and inventions are emerging every day, people are getting more dependent on machines to make decisions for them, it is essential that these people also understand the logic behind the decision-making process of a machine. With the assistance of explainable artificial intelligence, mankind could finally comprehend why or how a machine comes to a specific decision.

The reason why explainability and interpretability are important in artificial intelligence and machine learning lies in the fact that it was proven in the past how a machine learning could produce poor decision as a result of bias or discrimination without any intention of doing so. Especially black-box systems, they have been taking advantage of powerful machine learning process to produce predictions on individual information. In some cases, these predictions involve some rather sensitive and personal information, for example when it is about a credit risk assessment, credit score or insurance risks. Machine learning algorithms use a learning process based on digital traces from human's daily activities (online transactions, social networks activities, etc.) in order to produce predictions or make decisions. However, despite the enormous amount of data that people are producing every day, it is certain that they do not represent the behavior of every single human being. Therefore, the models that use this information as a learning base could automatically develop biases or prejudices, which results in unfair or wrongful decisions. As areas like healthcare and automobile applying artificial intelligence and deep learning system, the question of transparency and accountability is extremely important. If the algorithms of a system could not be interpreted and eventually explained, the potential of artificial intelligence would be highly limited.

In the topic of explainable artificial intelligence, the terms "explainable" and "interpretable" surface in every research paper. It appears that these two terms are closely related and, in most cases, could be used as synonyms, they share the same goal of making it understandable for human, why a machine would make a specific decision. However, there are some dissimilarities between the two terms in different aspects like technical aspect, human aspect and for different user groups.

On the other hand, this paper will describe methods for enhancing models' improvement, depending on which aspect, interpretability or explainability, different methods will be mentioned and analyzed. These are the methods and techniques which already surfaced

in other research papers and are developed by different experts. This paper simply tries to provide an overview as well as allocate them in the right group for their purposes.

2. Objectives

This paper is divided into two main parts, on one hand, the objective is to identify the differences between “explainable” and “interpretable” by definitions and comparations. Authors like Gall [32] or Doran et.al [25] have been trying to distinguish these two terms from each other, however, the aim is to dive deeper in this topic and clarify distinctive characteristics of each term.

The second part mainly focuses on different methods and techniques for model’s explainability/interpretability enhancement. In this part, it will be discussed the different models as well as methods and techniques for local or global explainability/interpretability as well as for explainable or interpretable systems.

Firstly, the idea of explainable artificial intelligence as well as its importance will be examined and demonstrated. After that, an attempt to define “explainability” and “interpretability” as well as their traits will be made. Aside from definition, each individual feature and the purpose of these terms will also be analyzed.

In the next part, it is important to analyze and evaluate the reliability of an AI system before trying to improve its explainability/interpretability. A system could be vulnerable to many aspects, from internal malfunctions to external attacks. Therefore, these aspects should all be taken into consideration while developing an AI system.

In addition to that, there have been numerous research papers on methods to improve model’s explainability/interpretability. However, as “explainability” and “interpretability” are seldom separated, therefore, these methods are often described in a general manner. Therefore, it is essential to draw a clear boundary between explainable and interpretable systems as well as methods.

Lastly, this paper analyzes the challenges as well as opportunities that XAI might encounter, some of which were already mentioned in the Problems section. However, in this section, the challenges and opportunities will be examined further as well as the important aspects one should pay attention to while implementing AI systems.

This paper should provide answers to these questions:

- What is explainable AI and why is it important?
- What distinguishes “explainability” from “interpretability”?

- Is there a difference in methodic implementation when it comes to the interpretability or explainability of the models?
- What are the challenges and opportunities for XAI?

At the end of this paper, it is expected that the meanings of “explainability” and “interpretability” are thoroughly inspected and separated. On the other hand, this paper should act as a guideline when it comes to choosing a suitable method to explain or interpret a model.

II. Analyzation

1. Explainable Artificial Intelligence (XAI)

1.1. Why XAI

Explainable Artificial Intelligence is a not new topic in the field of machine learning. The earliest research on this topic can be dated back to the 1980s where experts attempted to explain the results according to the applied rules [80]. As more researches on AI appear, there have been arguments on whether a system should be able to explain itself to the users, for example if a company applies artificial intelligence to help sorting out job applications, this system should be able to explain why or base on which criteria would a candidate be denied. This does not only provide the company itself of any biases the system might have, but also help the candidates understand the problems in their applications to avoid the same mistake in the future.

In general, a typical method with an explainable structure is a decision tree, an example can be seen in Figure 1 (Brid 2018). The process of a decision tree starts at the top then going down from level to level, the solution of a decision tree presents the reasoning of the final decision [94].

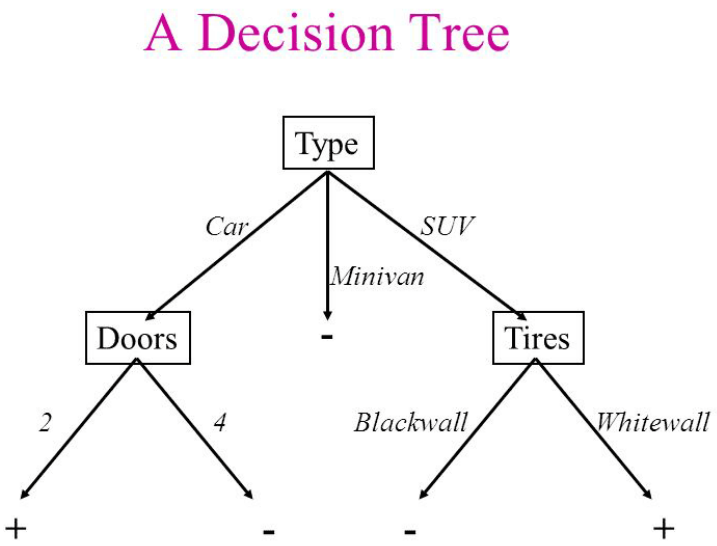

Figure 1: Example of a decision tree, start from the top then going down, level by level (Brid 2018)

However, according to Xu et al. [94], XAI has become a topic in context of modern deep learning. Most of the decisions made by these systems are processed through a black box, which could neither be explained by the system itself, nor the developer. It is clear that deep learning, in comparison to other structures, has been providing predictions with high accuracy, nevertheless, it is also proven that the better the performance, the lower the model explainability. As seen in Figure 2 from the DARPA Explainable Artificial Intelligence Program, explainable learning techniques like decision tree have the worst accuracy performance amongst others.

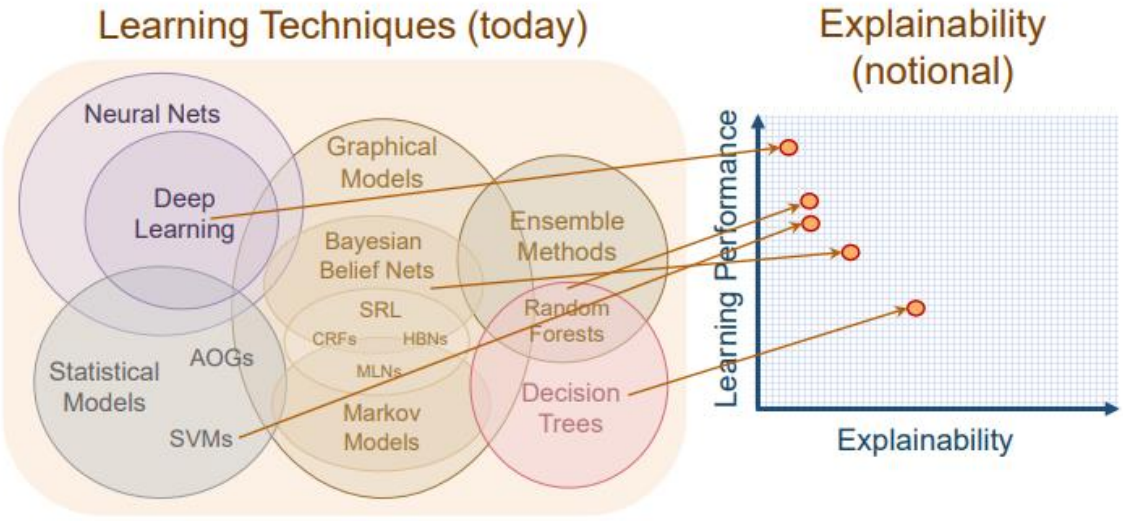

Figure 2: Relationship between explainability and performance of a model (DARPA 2017)

It raises the question, if the accuracy performance of deep learning systems is so high, why would there be a need for understanding the decision they make? As human being, it is important that one could explain the reasons behind every action that they make. The ability to justify the reason behind every decision is a significant aspect of human intelligence [78]. People tend to trust another person more if they understand the reasoning behind their decisions, while this aspect might not be as important for AI systems, there are several arguments supporting the necessity of explainable artificial intelligence. As Samek et al. [78] stated, here are the most important ones:

- **Verification of the system:** Despite the high accuracy of deep learning systems, one should not simply trust the decision made by black boxes. Especially in fields like medical, a system which could be verified and interpreted by experts is completely fundamental. An example for this was demonstrated by [4] when an AI system which was trained to predict pneumonia risk of a person has come to complete wrong conclusions. From the real data, this system has learned that an asthmatic person with heart problem has much lower risk of dying from pneumonia than a healthy person. A medical expert would immediately realize that this could not be true since asthma and heart problems are factors which negatively affect the recovery. In this example, the data which the AI model trained on was systematically biased, since the majority of asthma and heart patients are under strict medical supervision, they have significant lower risk of dying from pneumonia. However, this correlation does not represent the causal relationship between the two groups and therefore should not be used to support the decision on pneumonia therapy.

- **Improvement of the system:** Firstly, in order to improve a system, one must have the knowledge of its weaknesses. However, it is easier to perform such weakness analysis on explainable models than on black box models. On the other hand, if a model could explain how or why it has come to a decision, the biases such as in example above could also be simply detected. In addition to that, different models might have the same classification performance but largely differ in terms of what features they deem important for the decisions, a model interpretability would be useful for the comparison between these models. It is true that the more one understands what these systems are doing, even when they sometimes fail at their tasks, the easier it would be to improve them.

- **Learning from the system:** In the second game of the Go match between AlphaGo and Lee Sedol, AlphaGo has made a move which was deciding for this artificial intelligence to win the game against this top Go player. This move is described by Fan Hui (2016) as a non-human move. The AI system has developed new strategies to play Go, which has now certainly been adapted by human professionals. This proves that human could use AI systems to acquire new knowledge. These insights could be useful for domain such as sciences, since physicists, chemists and biologists are more interested in identifying the hidden laws of nature rather than just predicting some quantity with black boxes. Therefore, an explainable model would be of advantage in this case.
- **Compliance to legislation:** As technologies advance, artificial intelligence is getting more and more involved in every aspect of human's daily life, including legal aspect. The more power AI systems are getting, the more important it is to raise the question: "Who should take responsibility when an AI system makes the wrong decision?". However, it is almost impossible to find answer to this question relying only on black box models, thus, it is essential that AI systems in the future must become more explainable.

These arguments show exactly why it is important for AI systems to be able to explain themselves and an AI system could only be considered practical if it could provide an understanding of its mechanism or predictions.

1.2. Explainable Artificial Intelligence for different groups

Depending on who is asking the question, an explanation could differ from one target group to another. While everyone would like to receive a correct answer, the knowledge capacity of each group plays a massive role on how the answer might look like. Hind [43] stated that there are at least four distinct group of people who are interested in explainable AI with verifying motivations.

Group 1: AI system builders

Technical experts who develop and build AI systems such as data scientists or software developers. An understanding of the system would help them verify if the system is working as expected, how to debug the system in order to improve it. Moreover, it could also provide new insights from the system's decisions.

Group 2: End-user decision makers

These are people who rely on AI systems to help them make decisions, for example: managers, credit officers, judges or social workers. Since their decisions are affected according to the predictions of the AI systems, it is desirable that this group could come to understand the reasons behind each prediction in order to improve trust and confidence in the system's recommendations. Furthermore, it is possible for these users to acquire knowledge and improve future decisions by understanding how a system comes to a prediction.

Group 3: Regulatory bodies

Authorities and governments would like to ensure that every decision is made in a safe and fair manner and the society is not negatively impacted by decisions such as financial crisis.

Group 4: End consumers

Since this group is directly affected by decisions made by AI systems, it is just fair that they acquire proper reasoning for the decisions which have been made about them.

As mentioned above, the knowledge capacities of each group have direct impact on how the answers would look like. However, every person who works with AI systems will need a better understanding of what the system is doing.

Depending on who is asking the question, an explanation could differ from one target group to another. While everyone would like to receive a correct answer, the knowledge capacity of each group plays a massive role on how the answer might look like. Hind [43] stated that there are at least four distinct group of people who are interested in explainable AI with verifying motivations.

2. Explainability

Defining what an explanation is, is the starting point for creating explainable models, and allows to set the three pillars on which explanations are built: goals of an explanation, content of an explanation, and types of explanations [72]

2.1. What is an explanation?

The following definitions can be found for the word "explain": "to make something clear or easy to understand by describing or giving information about it" [28]; or "to tell somebody about something in a way that makes it easy to understand" [29]. In other word, to explain something is to make something understandable for the other person.

On the other hand, Lewis [49] states that “to explain an event is to provide some information about its causal history. In an act of explaining, someone who is in possession of some information about the causal history of some event – explanatory information – tries to convey it to someone else”.

According to Halpern and Pearl [38], a good explanation which provides the answer to the question “Why?” is defined as (A) “provide the information that goes beyond the knowledge of the individual asking the question” and (B) “be such that the individual can see that it would, if true, be a cause of”

There is obviously no consistency in defining explanation, however, there are some similarities in these definitions. Firstly, the goal of an explanation is to give the individual raising the question an understanding about something. Secondly, an explanation often relates to the question “why” or causality reasonings.

Explanations are required in many fields and this concept probably exists since the beginning of human communication. Proofs are normally used by mathematicians to formally provide explanations. The logic used for these proofs are agreed-upon, so that every other mathematician could verify whether these explanations are valid. However, these agreed-upon formalisms do not apply for non-mathematical explanations. Providing a satisfying explanation from human to human is already difficult, expecting to get one from a system is a real challenge. Even though we do not have a specific set of criteria to determine when an explanation from a system is enough, it cannot be denied that an AI which can explain itself is essential.

2.2. Characteristics of explanation

Molnar [57] mentioned different characteristics of explanations in his book about interpretable artificial intelligence. According to the author, there are properties which could be used to evaluate how good an explanation method or an explanation is. However, how to measure these properties correctly remains a mystery, therefore, one of the challenges is to formalize how they could be calculated.

Properties of explanation method:

- **Expressive Power:** defines the structure of the explanations the method is able to generate. An explanation method could generate IF-THEN rules, decision trees, a weighted sum or something else.

- **Translucency:** describes how much the explanation method relies on looking into the machine learning model, like its parameters. For example, methods which only manipulate the inputs and observe the predictions have zero translucency. The level of translucency might vary depending on different situations. The higher the level of translucency, the more information the explanation method can extract in order to provide explanations. However, low translucency means the method is more portable.

- **Portability:** describes the range of machine learning models with which the explanation method can be used. Low translucency level means that the method is more portable since it is treating the whole model as a black box.

- **Algorithmic Complexity:** defines the computational complexity of the method that generates the explanation. It is important to take this property into consideration when computation time is a bottleneck in generating explanations.

Properties of individual explanation:

- **Accuracy:** describes how well the explanation predicts unseen data. High accuracy is important if the explanation is used for predictions in place of the machine learning model. It is fine to have low accuracy if the accuracy of the model is also low, and if the goal is to understand what the black box does. In this case, only fidelity matters.

- **Fidelity:** describes how well the explanation approximate the prediction of the black box model. High fidelity is one of the most important properties of an explanation, since an explanation with low fidelity is useless to a machine learning model.

- **Consistency:** describes how much the explanation differs while being used on different models which were trained on the same task and produce similar predictions. If an explanation method produces similar explanations on models with similar predictions, it is highly consistent. However, high consistency does not always guarantee a good explanation. For example, two models could produce similar predictions but using different features. In this case, high consistency is not desired since the explanations differ from each other. Thus, high consistency is only useful when the models rely on similar relationships.

- **Stability:** describes how much the explanation differs while being used on similar instances in the same model. High stability is always desirable, since high stability means that slight variations in the features of an instance do not change the explanation, except when these variations also change the prediction.

- **Comprehensibility:** defines how well human could understand the explanation. This feature is difficult to define and measure, however, it is important to get it right. Measuring comprehensibility includes measuring size of the explanation or how well people can predict the behavior of the machine learning model from the explanations.
- **Certainty:** defines whether the explanation reflect the certainty of the machine learning model. An explanation with model certainty is very useful since it reflects the model's confidence about how correct the prediction is.
- **Degree of Importance:** describes how well the explanation reflect the importance of features or parts of the explanation.
- **Novelty:** The concept of novelty is closely related to the concept of certainty. The higher the novelty, the more likely that the model has low certainty due to lack of data.
- **Representativeness:** shows how many instances an explanation covers, it could either represent the entire model or just an individual prediction.

2.3. Explainability in Machine Learning

A model is explainable when it is giving the users a proper explanation for its predictions. These explanations could be in words or visualization which help the users connect the input with the output. Depending on how much knowledge the users have on the topic, it is his responsibility to put these explanations together and understand them.

Explainablity is an active characteristic of a model, indicating which steps the model has taken in order to clarify or provide details about its internal mechanism [4]. Gall [32] shared the same opinion, however, he added that these details about internal functions of a system should be explained in human terms.

As can be seen in Figure 3, a model is explaining why it would predict that a person is having a flu. LIME (Local Interpretable Model-Agnostic Explanations) is being applied here as an explainer which shows that symptoms like sneeze or headache are treated as relevant to flu while no fatigue is not. From this explanation a human doctor could personally decide whether the prediction was right or wrong. In this case, it is obvious that a human doctor is better qualified to make the final decision considering that the system provides reasonable explanation for its recommendation. It is simpler for human with prior knowledge to trust a system's prediction when proper reasonings are presented.

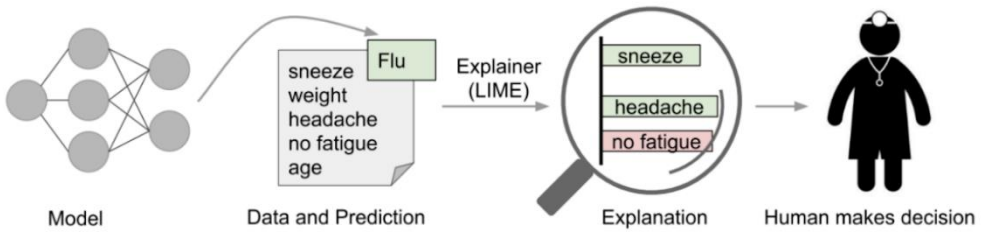

Figure 3: Explaining individual predictions (Ribeiro 2016)

Explainability includes the techniques to convert a non-interpretable model into an explainable one, therefore it is connected to post-hoc explainability [27]. Post-hoc systems focuses on explaining a specific system locally and make it possible to reconstruct the system if necessary [44]. Post-hoc analysis is used as an assistance in explaining reasons for complex models such as deep neural network or deep forests [11]. In this analysis, the users tend to try and explain the prediction of a model after seeing the output data.

In other word, explainability is about extracting information from a model. It might not show concretely how a model works internally, however; it provides the users with useful information. By applying explainability, it is possible to explain opaque models without the risk of losing predictive accuracy. There are three main approaches to post-hoc explainability: language explanations, visualizations of learned models and explanations by example [50].

The goal of language explanations is to provide explainability for a model by creating explanations in words which make the decisions of the model understandable for users. These explanations consist of all the symbols that demonstrate how a model operate, they represent the logic of the algorithm through semantic model-symbols mapping. In addition to that, according to Ribeiro et at. [70], natural language explanation consists of verbal words and visualizations which “provide qualitative understanding of the relationship between features of an input (e.g. words in a documents) and the model’s output (e.g. classification or prediction). As humans explain themselves with words, one model will be created exclusively with the purpose of providing predictions and another model will be used to generate explanations, for example a recurrent neural network (RNN) language model.

Visualization aims for visualizing the model’s behavior. This could be achieved through dimensionality reduction techniques that use simple visualization to make it understandable for human. It is also possible to combine visualization explanations with other techniques to enhance the understanding of the model. This technique is considered the most suitable method which helps the users understand the variables in a complex

model with limited knowledge about machine learning. Visualizations is also about representing through images how different features or pixels influence the system's prediction or provide the users with an interface so that they can analyze the explanations. t-Distributed Stochastic Neighbor Embedding (t-SNE) is one of the most popular approaches in this method which provides each datapoint a location in a two or three-dimensional map in order to create a high-dimensional data visualization [52].

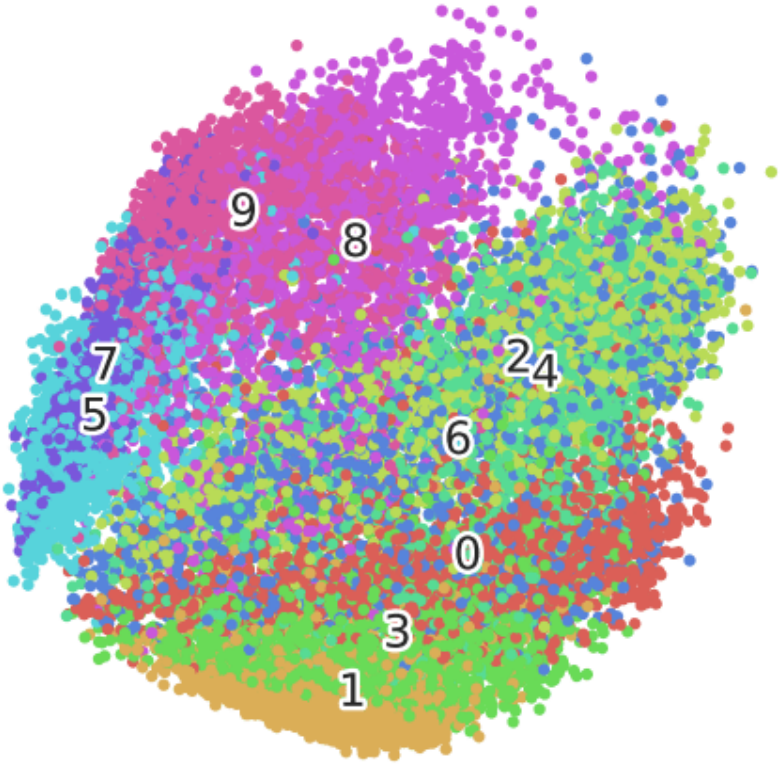

Figure 4: Visualization of MNIST-Fashion Dataset using PCA (Pathak 2018)

Figure 4 shows the linear feature extraction technique called Principal Component Analysis (PCA), which performs a linear mapping of the data to a low-dimensional space. In comparison to PCA, t-SNE presents a much better visualization of the clusters and the hidden structures of the dataset, which can be seen in Figure 5.

Lastly, explanation by example is a post-hoc mechanism in which the system attempts to extract data examples which are considered most similar to the predictions that were generated by the model. These examples could then be shared with people who are affected by the predictions. This technique focuses on the representative examples that show the users the inner relationships and correlations created by the model [4].

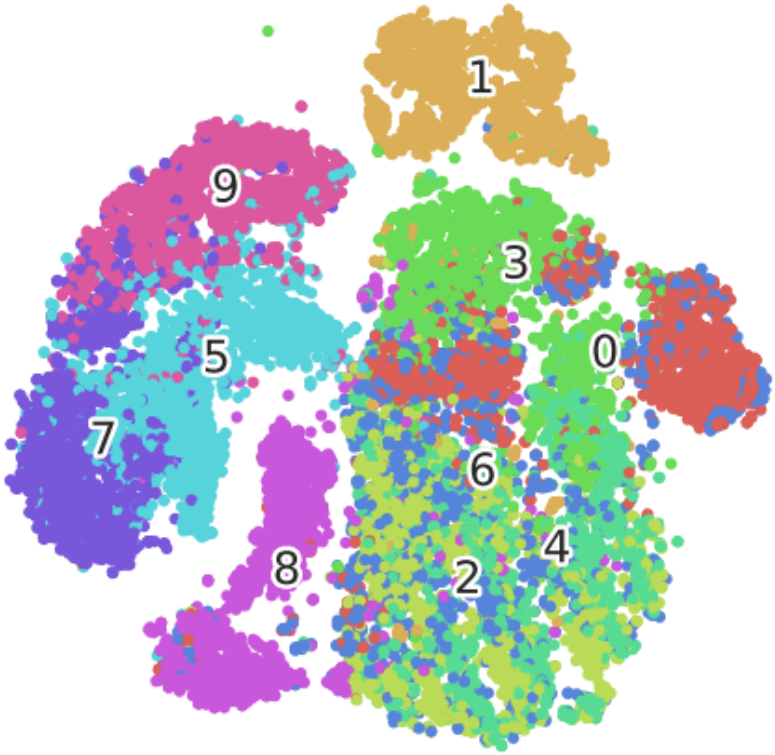

Figure 5: Visualization of MNIST-Fashion Dataset using t-SNE (Pathak 2018)

In the paper published by Arrieta et al. [4], three extra techniques are presented to post-hoc methods which help enhancing model's explainability, which are local explanations, explanations by simplification and feature relevance explanation.

Local explanations describe the explainability of a model is approached through solution space segmentation. Explanations are generated for less complex solution subspaces; however, these subspaces must be relevant for the whole model. To generate these explanations, one could use techniques with non-identical features to explain part of the whole system's functioning.

Explanations by simplification requires a simpler implementation of a whole new system which is built according to the trained model that needs to be explained. This new model normally tries to optimize the similar functioning of its prior model in a less complicated art but still maintain the performance score. Furthermore, it is generally easier to implement this model since it is less complex in comparison to the model it represents.

Lastly, feature relevance explanation technique is an indirect method to explain a model. It shows the inner functioning of a model by identifying the relevance score for its variables. These scores demonstrate the sensitivity a feature has upon the output of the model. How each of these variables affects the output of the model will be evaluated based on the comparison of their relevance scores, the higher the scores, the more important the variables will be for the model.

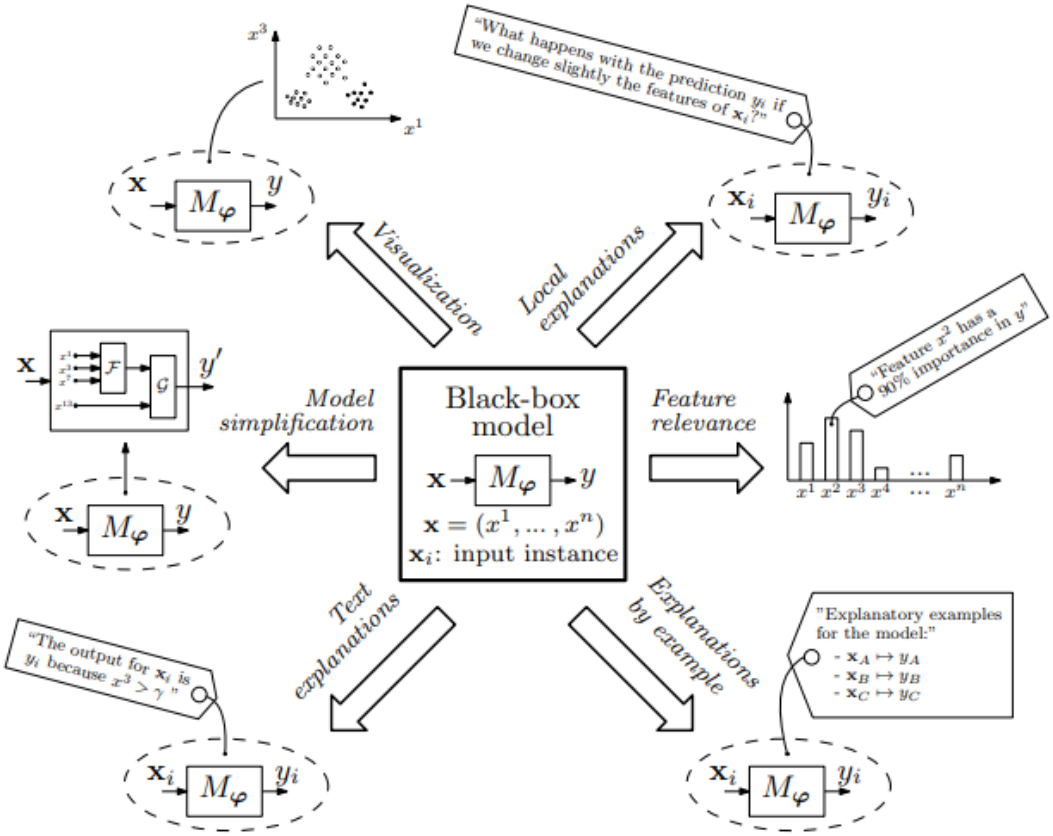

Figure 6: Conceptual diagram of different post-hoc methods for improving explainability in machine learning models (Adadi et al. 2019)

3. Interpretability

3.1. Interpretation

To interpret something means “to explain the meaning of something” [29] or “to decide what the intended meaning of something is” [28]. Like explanation, it is difficult to define interpretation in one sentence, this term may vary depending on the one who is trying to achieve an interpretation for an event. Glasersfeld [34] explained that the concepts, sensations, thoughts or emotions of one person can never be compared with another’s, which means one person could interpret something in one way meanwhile another could see it in a completely different light.

On the other hand, in order to create an interpretation for something, one must at least have experience on that topic. It is impossible to get an accurate interpretation from the start, a correct interpretation often relates to a series of trials, errors, and different variations.

In the history of literature, the act of trying to comprehend what an author meant through a poem, for example, would be considered interpretation. However, one poem could be interpreted in many ways depending on the interpreter. It is also possible that none of these interpretations match the one that the author had in mind.

3.2. Evaluation of interpretability

There are three main levels for the evaluation of interpretability proposed by Doshi-Velez and Kim [26]:

- **Application level evaluation (real task):** The explanation will be put directly in the product and tested by end user. This involves real human experimenting with real applications. For instance, if there is a system which connects students with similar learning goals, this system will be tested by the students and the result will be whether the students achieve better grades with the assistance of the system. However, it is important for human to have a good explanation for the same decision since this is the baseline of this level of evaluation.
- **Human level evaluation (simple task):** This is a simpler version of application level evaluation. The main difference in this level is that the people taking part in experiments are non-experts, which makes these experiments cheaper and the participants are easier to find. There are different approaches to these experiments: binary forced choice (choosing explanation with better quality out of two), forward simulation/prediction (predicting the output based on the input and explanation) or counterfactual simulation (trying to achieve desired output having known the input, explanation and the output).
- **Function level evaluation (proxy task):** In this level of evaluation there is no human required, alternatively some formal definitions of interpretability will be used to represent the quality of the explanation. The best-case scenario would be when the class of the model has already been validated beforehand, for example in a human level evaluation. This kind of experiments is appropriate in case the method is not mature or the experiments on human might be unethical.

3.3. Interpretability in machine learning

Interpretability can be considered as a tool of measurement for experts with deep understanding of the system to make decision about the model. By applying interpretability in machine learning, experts could gain such knowledge, which enable them to build the system, maintain it or debug it, as well as operate it.

Interpretability describes the technical side of a system, most of the time interpretable AI tries to make sense of transparent systems, which means every aspect of the system is visible. Generally, a complete transparent system would be unrealistic, it would be expected that there are different degrees of transparency. Ronan et al. [74] proposed three levels of transparency in AI systems:

- **Implementation:** This is a level where the relation between input and output data is clear, this consists of technical principles and the corresponding parameters. This level of transparency is standard for most open source models and is often referred to as white-box models.
- **Specifications:** This level provides all information that are needed in order to obtain the implementation, which includes details about model training dataset, training procedure and performances.
- **Interpretability:** This refers to how the model gives reasons for its prediction; it shows that the model follows the specifications and support human values. However, most of the AI systems currently in use do not possess this level of transparency.

Transparency leads to interpretability and is also considered a logical first step to protect human-based institutions. Moreover, transparency is now translated into a prime solution to algorithmic concerns such as discrimination or biases [27]. However, as technologies develop, AI models have become much more complex than just human-based institutions and it is getting more difficult to achieve an understandable explanation for users. Lipton [50] defined transparency as:

“Informally, transparency is the opposite of opacity or blackbox-ness. It connotes some sense of understanding the mechanism by which the model works. We consider transparency at the level of the entire model (simulatability), at the level of individual components (e.g. parameters) (decomposability), and at the level of the training algorithm (algorithmic transparency).”

Simulatability: If it is possible for a person, for whom the predictions are intended, to internally simulate and explain the whole process of decision-making of a system, then the system can be considered simulatable [70]. A model could only be fully understood if a human could connect the input data with the parameters of the model and proceed through all the necessary steps to achieve the prediction in reasonable time [50]. However, a simulatable model at the same time must be simple since human does not possess the same calculating capacity as machines. Examples for simple models are decision trees or lists of rules, which are easy to simulate. On the other hand, these models often have high

descriptive accuracy because of their simplicity, and high descriptive accuracy means highly effective models.

Decomposability: Another aspect of transparency is when each feature of a model acknowledges an intuitive explanation [50]. Each node of a decision tree must have a description to it, or a linear model has the parameters which represent how much a feature is related to a label. It is important to keep in mind that individually interpretable inputs are required for this aspect of interpretability. However, despite the possible intuitive weights of a linear model, they could still be vulnerable depending on feature selection or pre-processing. This is the reason why one should not simply trust a model only based on its popularity.

Algorithmic transparency: This means all factors which are directly involved in the prediction of the system must be transparent to the people who are using the system or affected by the system. However, in neural network model, it is impossible to achieve algorithmic transparency since all the algorithms are happening inside a black box.

4. System reliability in AI

The performances of AI systems have been proven to be powerful in recent years, however, it is not possible to completely rely solely on their prediction. Any complex decision based on an AI system should be under strict human supervision since it could lead to undesirable consequences. For example, the first autonomous car accident was recorded in Arizona in 2018, where an Uber-operated car struck and killed a pedestrian, this happened even though there was an emergency back-up driver behind the wheel. According to the National Transportation Safety Board report [25], the reason for this accident was that shortly before the impact, the system predicted the pedestrian with a bicycle as a vehicle and unknown object. The system was built without the consideration of jaywalking people, therefore, by the time the system recognized the risk of collision it was already too late for the driver to intervene.

Poor performances and vulnerabilities are the signs showing that a machine learning model is not reliable. According to Ronan et. al. [73], in order to assess the reliability of a system, it is necessary to look at these two aspects.

4.1. Evaluation of performances

Evaluation of performances is a crucial part in the process of developing a machine learning system. It consists of many questions including choosing the right metrics and the procedure of evaluation.

How well a system could solve a problem depends a lot the right metric choice. Normally, it is determined by the sort of data, the choice of class of models, and the task specifications. Regardless of how correct the chosen metric is, it will still be just a guess and therefore it would not include all the parts of the examined task.

Methods such as create a training set and a test set from the dataset or taking the imbalance of the dataset into consideration have been applied commonly in machine learning for evaluation procedure. However, there are some aspects relating to reliability which should also be taken into consideration [69].

Firstly, internal validation is used for evaluating the performance of a diagnostic or predictive model, on the other hand, external validation is used to evaluate the performance with data which is not relevant to the development of the model. Since internal validation is a method using previously asked questions to test the model, it could eventually misjudge the model's performance. Therefore, external validation is extremely important in case of overparameterized diagnostic or predictive image classification model which contains high-dimensional data. External validation prevents overfitting where a model gets too familiar with the training dataset and produces negative effects on model's performance.

Secondly, another aspect is the risk of spectrum bias, which according to Ronan et al., "refers to the presence of examples in the dataset that does not reflect the diversity and complexity of situations, i.e., the spectrum of examples does not reflect the real spectrum". This statement shows that great performance on examples which are obvious does not mean that the model is going to perform accurately when it comes to more complex situations.

The last aspect that was mentioned in [69] was regardless of the outstanding performance, there is always a risk that a system might provide predictions which are not useful or failures in operating artificial intelligence systems while trusting their predictions blindly.

4.2. Vulnerabilities

AI systems are disposed to more vulnerabilities than traditional systems. Knowledge of the parameters of the model is normally required to make use of these vulnerabilities or directly manipulate how the models work, however, it cannot be ruled out that the system is safe from external attacks. The most common vulnerabilities relating AI systems include data poisoning, crafting of adversarial examples and model flaws [73].

Data poisoning: Attackers could establish false training data on purpose which could neutralize the system or lower the performance. The ability of a model to learn new patterns by continuous retraining in real time using newly obtained data contributes to data poisoning. With this, attackers could feed the model incorrect data which could lead to wrong predictions or maximize the wrong goals in reinforcement learning systems. Data poisoning could also happen when an attacker has access to the training data. Due to the complexity of AI systems nowadays, it requires a massive amount of computational and human resources to train a model. Since it is not rare to reuse a model that has already been trained by someone else, it creates an opportunity for adversaries to attack and manipulate the models.

Crafting of adversarial examples: Molnar [57] defines an adversarial example as “an instance with small, intentional feature perturbations that cause a machine learning model to make a false prediction”. Machine learning systems are more vulnerable to attacks due to adversarial examples, for instance, an autonomous car could get in an accident when a stop sign is manipulated to look like a parking prohibition sign for the sign recognition software. Adversarial examples are often created with pixels which were intentionally perturbed to deceive the model while it is being operated. These pixels are often incomprehensible for human eyes; however, they could easily trick a system into recognizing false objects, in another word, adversarial examples are optical illusions for machines [57].

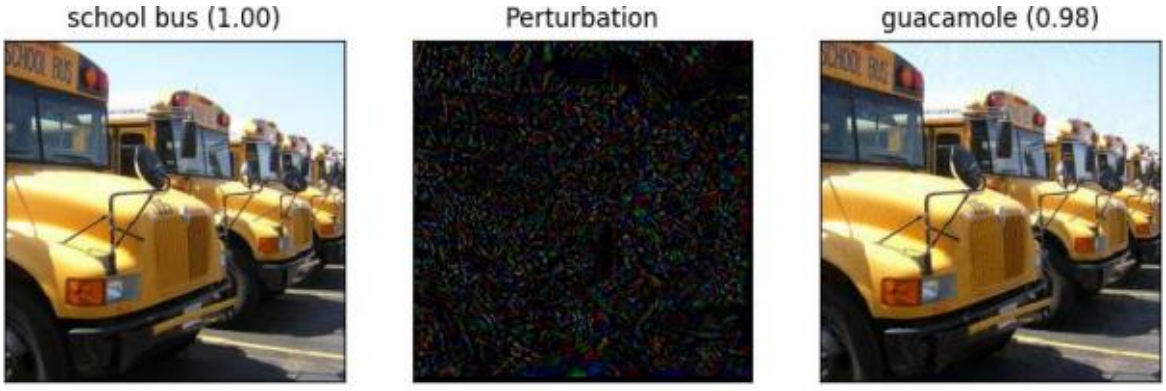

Figure 7: Adversarial example using Basic Iterative Method (Ronan et al.)

Figure 6 shows how an adversarial example might look like in the context of image classification. On the right hand side, the model correctly recognize the school bus in the picture; nevertheless, after applying perturbed pixel image, which looks totally harmless to human, the model ended up falsely predicted the picture as guacamole with high confidence. Image classification is not the only field that is vulnerable to adversarial examples, Ronan et al. [73] also stated that other contexts such as image segmentation, object recognition, speech recognition, text summarization and summarized or generative models could also become victims of adversarial examples.

Model flaws: Existing weaknesses of the mathematical procedure within the learning process of the model can be exploited by adversaries. Using specific architecture is often more open to vulnerabilities, for example the existence of noise could be used by attackers to trick the system. These attacks are normally happening in unreliable settings, however, considering how easily it is to create them in limited context, having these vulnerabilities in real-life situations could lead to fatal consequences

4.3. Increasing AI model reliability

Based on different kinds of vulnerabilities, attack intention and the model type, there are different approaches in order to prevent a model from operating unexpectedly: data sanitization, robust learning, extensive testing and formal verification [73].

Data sanitization: Training data should free of all potentially malicious content before getting trained in the system to avoid data poisoning. This could be done by implementing another AI system to filter or defining rules for classical input sanitization. However, when the situation is important, human intervention is required.

Robust learning: Learning procedure could be altered to become robust against attacks, especially in case of adversarial examples as mentioned above. To be able to achieve that level of robustness, systems are trained to recognized known adversarial examples, moreover, mathematical foundation of the algorithms will also be redesigned through statistical techniques (regularization and robust inference).

Extensive testing: One training dataset is normally limited and does not cover every aspect that could occur in real life. Therefore, a model should be tested for different circumstances to ensure its reliability in case a specific example has not been covered in the training dataset. Figure 7 shows 15 different types of corruptions in four main categories which can appear on censors: noise, blur, weather and digital. This helps increase not only the reliability of a system, but also its consistency and stability in case of minor input changes [41].

Formal verification: Formal verification has the goal of proving whether a software or hardware system is correct with the assistance of mathematical proofs, based on specified properties. There are two main properties that need to be considered: (Un)satisfiability and robustness. (Un)satisfiability checks whether it is possible to get a certain output from a specific input. Robustness checks whether the output will be changed if noise is added to a given input. For example, while analyzing a malware, it is plausible that malicious contents should be recognized and should not be considered harmless. However, it is impossible to verify all malicious contents through input and there is no specific definition what a malicious file could contain. Therefore, by using and modifying existing data while keeping the malicious part of the data, it could be possible for the model to classify malicious files correctly.

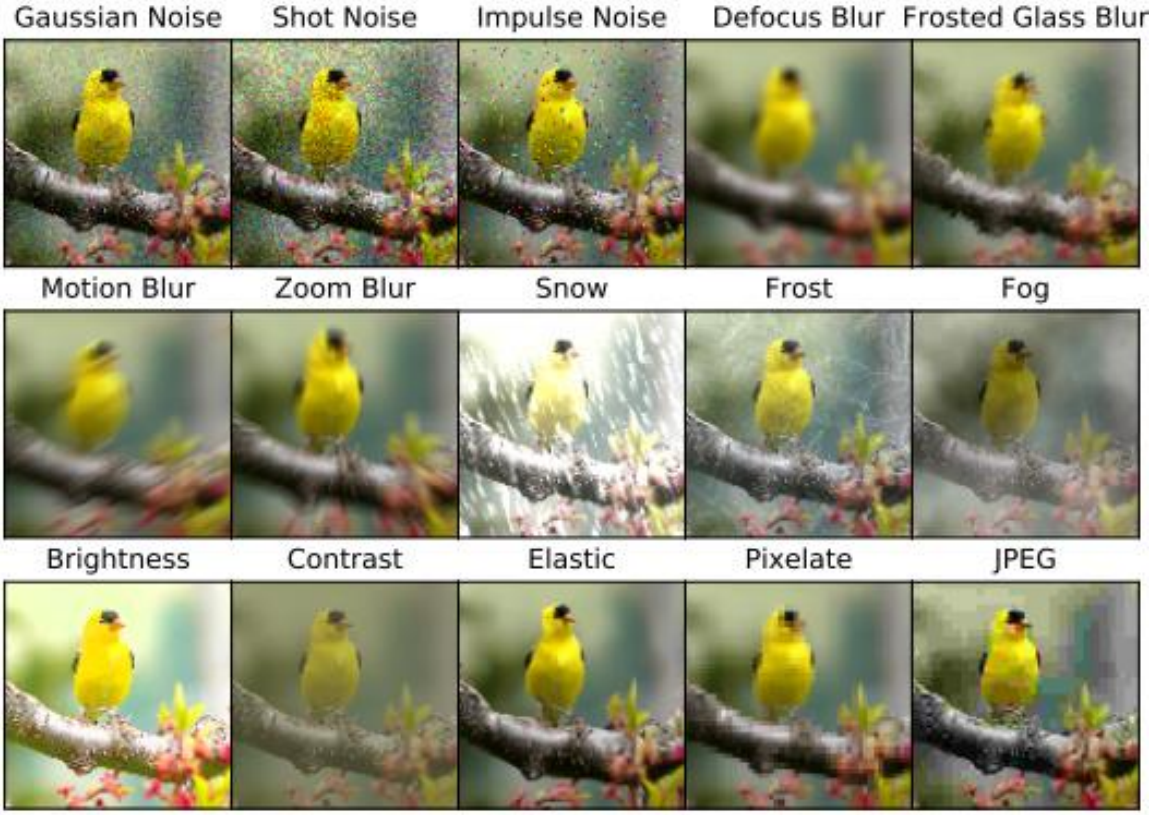

Figure 8: 15 types of algorithmically generated corruptions from noise, weather, and digital categories on a picture from ImageNet dataset (Hendrycks et al. 2019)

Nevertheless, these methods often work well on linear models since the existing algorithms are efficient and scalable. Complex machine learning algorithms are normally non-linear, such as neural network and these algorithms do not apply to them. There are several techniques for non-linear model (approximations of non-linear functions [46], propagating regions around inputs [45] or clustering inputs [36]), however, even though they produce promising results, they are not suitable to apply in large network.

5. Methods for improving explainability/interpretability in machine learning models

Figure 9 shows the different types of explainability/interpretability methods for modern machine learning models. The extent of explainability/interpretability in a model could either be global or local, meanwhile, depending on the purpose of a model, whether it should be explainable or interpretable, post-hoc methods or transparency could be applied to support this. Basically, as written in the previous parts of this paper, the main difference in modern machine learning/AI models is between true transparency (interpretability) or post-hoc explainability.

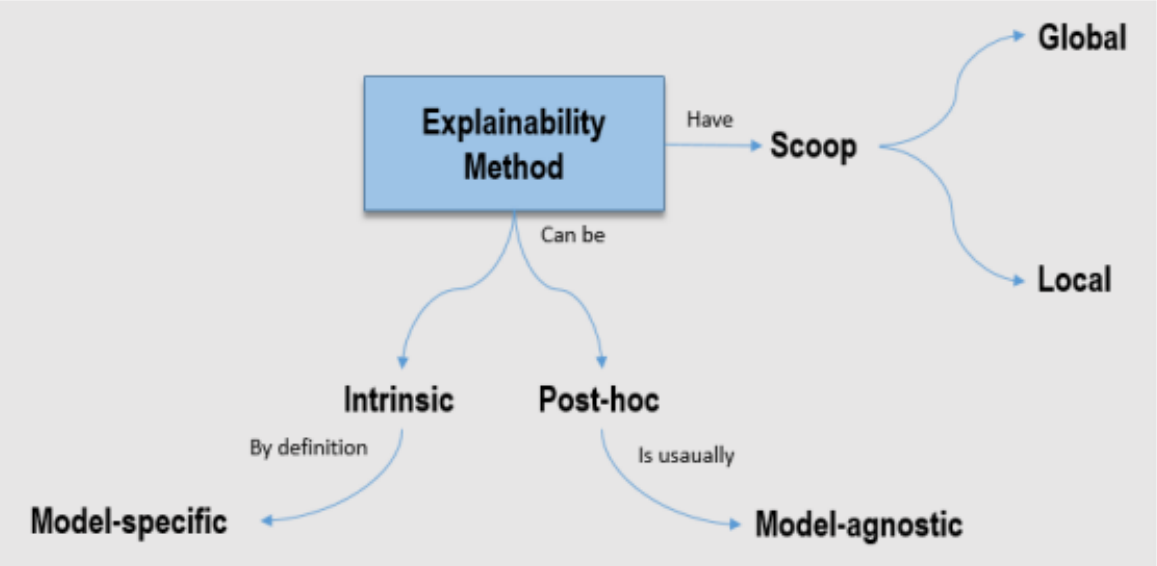

Figure 9: A pseudo ontology of XAI methods taxonomy (Adadi et al. 2018)

5.1. Global interpretability/explainability

Global interpretability/explainability is the state where the users have complete knowledge of how a complex model works globally by examining its structures and parameters. It indicates when the logic of a model is understood as a whole, and the users could understand every reasoning that leads the model to the final decision or prediction [1].

A simple but effective global model interpretation was proposed on the published paper of Yang et al. [95] where she used recursive partitioning to create a compact binary tree which represents the most important rules which contribute to the decision of the black-box model. The author also stated that this method could determine whether a model is behaving reasonably or overfit to some unreasonable pattern.

Another method was proposed by Valenzuela-Escárcega et al. [85] called entity classification. This method is based on a lightly supervised approach to information extraction, which produces a global, deterministic interpretation. In this method, a combination of traditional bootstrapping which the author refers to as “use of limited annotations and interpretability of extraction patterns”, and representation learning is applied to get the benefit out of both.

Activation maximization is also another method for global interpretation proposed by Nguyen et al. [61], which synthesizes an input which highly activates a neuron while using a deep generator network as a learn prior: Deep Generator Network. Human brains are proven to react to specific, abstract concepts such as Halle Berry or Bill Clinton, similarly, scientists aim to identify factors which highly stimulate a neuron by documenting a neuron’s activation when shown different images. Nguyen et al. takes advantage of DGN to create an algorithm which not only could create images as realistic as possible, reveal the features learned by a neuron in an interpretable way, operate well with new datasets and to an extent, even new network architectures without the necessity of relearning the DGN, but also would be considered a high-quality generative method for producing realistic, interesting, creative and recognizable pictures [61].

Despite having many researchers working on different methods for achieving global interpretation, it seems unlikely that this could be successful in practice, especially when a system is so complex with many different parameters. In comparison to global interpretability, a local explanation would seem more likely to yield promising results.

5.2. Local interpretability/explainability

Adadi et al. [1] defines local explanations as an attempt to approach interpretability/explainability by understanding reasons for specific decisions or a single prediction instead of trying to make sense of the whole model. These are several methods exploring local interpretability/explainability that have been mentioned in different papers.

An example from the previous part of this paper has mentioned LIME (Local Interpretable Model-Agnostic Explanations) method presented by Ribeiro et al. [70]. The name of the method already reveals that this is a method suitable only for local explanations. This method aims to achieve an explanation which could be represent in an understandable way and is locally faithful to the classifier.

Another method for local explanation was proposed by Baehrens et al. [7], where he uses a framework with local explanation vectors, which could be applied to any classification method to produce an understandable reason for single data instances. Based on this research, other methods are developed especially for image classification models, for example saliency maps, sensitivity maps, or pixel attribution maps. These methods share the same trait where they use techniques with gradients to assign an “importance” value to individual pixels which in the end reflect the influence on the final decision.

In a more recent work, Lundberg and Lee [51] have proposed a newer technique called SHAP (Shapley Additive exPlanations), which aim to unify all local approaches. The goal of SHAP is to assign each feature an importance value for specific decision. A more detailed explanation on this method will be presented in the later part of this paper.

5.3. Post-hoc explainability techniques

Post-hoc explainability is normally suitable for models which are not interpretable by design (black-box models) and uses different techniques with the aim to improve the models’ explainability. As mentioned above, these techniques include text explanations, visual explanations, local explanations, explanations by example, explanations by simplification and feature relevance explanations [4]. Post-hoc explainability makes use of model-agnostic methods, since they are model-independent, they could be applied to any machine learning model, regardless of its inner processing or representations [1]. Model-agnostic methods may depend on explanations by simplification, feature relevance or visual explanations [4], which will be discussed further in this section.

5.3.1. Explanations by simplification

Explanations by simplification are without a doubt the most wide-ranging technique amongst all model-agnostic post-hoc methods. One of the most popular approach is LIME [70], including all its variations. This method creates locally linear models around the predictions of an incomprehensible model so that the model could be explained. LIME aims to provide an understandable explanation for any black-box model [37]. Moreover, since the models have less complexity and are built locally, this method belongs to both explanations by simplification and local explanations. Another technique which was also based on rule extraction is G-REX, where a reverse engineering schema is applied to annotate random permutations of a dataset and the dataset is used as input by G-REX. To be more precise, G-REX exploit genetic programming as a key concept to extract rules [37].

In accordance with rule extraction methods, Su et al. [82] proposed different approach to learn two-level Boolean rules in CNF (Conjuctive Normal Form) or DNF (Disjunctive Normal Form) for human-interpretable classification models. This approach's objective is to find the middle ground between the accuracy of classification and the simplicity of the rule.

Following the same path, in [10], the authors presented a model extraction approach for complex black-box models' explanation. This method is somewhat a mixture of interpretability and explainabilty since a transparent model is used to approximate the complex one. However, it falls under the category of explainability, considering that the transparent model is only developed with the aim to explain the main model.

Tan et al. [83] worked on another technique for explanation by simplication called "Distill-and-Compare" (Figure 10), which revises a black-box model without having to examine the black-box API or pre-defined functions. The black-box model is hereby treated like a teacher and the students are the transparent models, which are trained to imitate the black-box risk scoring model's behavior. In addition to that, another model will be trained to predict outcomes based on audit data. Differences between feature regions of the two models will be examined to understand what could have happened inside the black-box model that makes the mimic model different from the outcome model. In the end, a statistical test will be applied in order to check whether there are any other features that the black-box model was using, which the authors have no access to [83].

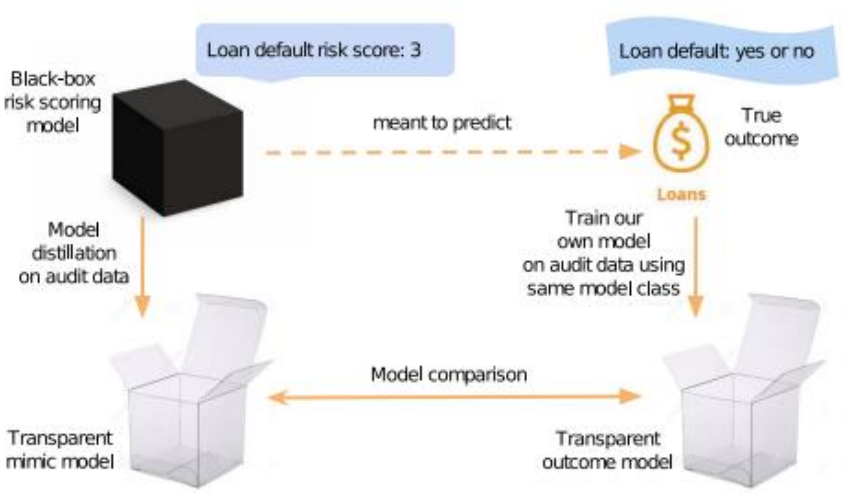

Figure 10: Distill-and-Compare approach on loan risk scoring model (Tan et al. 2018)

Considering the amount of recent research which focuses on explanation by simplification approach, this proves that this post-hoc explainability method is getting more popular and will become a central part in the development of explainable artificial intelligence.

5.3.2. Feature relevance explanation

Feature relevance explanation has the goal of making opaque models comprehensible by describing their functioning. This is done through ranking or measuring how relevant or important or how much influence each feature has on the outcome prediction of the model that needs to be explained. There are different algorithms to this approach even though they all aim for the same goal. One of the techniques that delivers most potential results is SHAP, which was already mentioned in the previous part of this paper. SHAP was proposed by Lundberg et al. as “a unified measure of feature importance”.

Before getting to know SHAP, it is important to have an idea what Shapley values are. Shapley value applies classic equations from cooperative game theory in order to attribute the payouts of the games fairly and axiomatically between players. There are four traits which are supposed to be fulfilled [90]:

- **Efficiency:** The profit equals the sum of all bonuses
- **Null Player:** If a player does not contribute to the game, he does not get a payout
- **Symmetry:** Players which are in similar situations get equal payout
- **Additivity:** If player contribute to two different coalition functions, the total payout he receives should be equal to the sum of the payout that he would receive from each function separately

Shapley value is the only method that satisfies all these characteristics. In machine learning model, each input will be allocated to a certain degree of importance which affects the final prediction (Figure 11). The Shapley value is calculated by getting the average of all the marginal contributions to every possible scenario that could happen which would attribute to the output [57]. Using Shapley explanation as an additive feature method, Lundberg et al. has created a combination of LIME and Shapley values. There are three properties which are expected to be fulfilled by using SHAP: local accuracy, missingness and consistency.

Koh et al. also proposed another approach to trace back the model’s prediction to the training data which is called influence functions [47]. This is a classic technique from robust statistics, by going through the learning process of a model all the way back to the training data, it is possible to identify the training points that have the most influence on a given prediction. This technique has been proven to be useful on both linear models and convolutional neural network when it comes to “understanding model behavior, debugging models, detecting dataset errors or creating visually distinguishable training-set attacks” [47].

There are also many other papers on the topic of feature relevance explanation, such as coalition game theory [81], local gradients [72], QII (Quantitative Input Influence) [21] or ASTRID method (Automatic STRucture IDentification) [42]. Considering the amount of research which surfaced recently on this topic, it is undeniable that feature relevance explanation technique is getting more attention and popularity amongst experts who share the goal of tackling the field of explainable artificial intelligence.

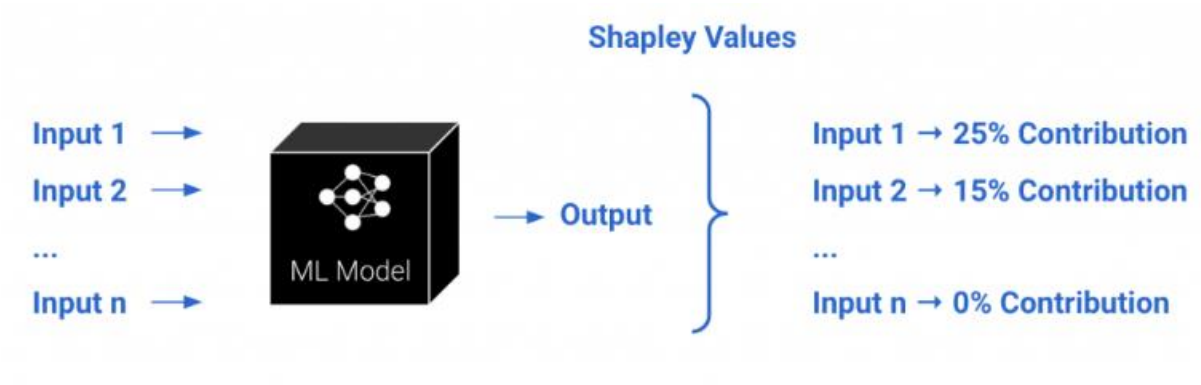

Figure 11: Shapley values applied to ML models (Taly 2020)

5.3.3. Visual explanation techniques

Visual explanation techniques could be considered a pathway towards model-agnostic explanations. Based on Sensitivity Approach (SA), Cortez et al. [20] has developed a method to open black-box data mining models. The authors proposed a Global Sensitivity Approach (GSA), which combines traditional SA methods and some visualization techniques to assign input relevance and how it is responsible for the model's prediction. SA was initially developed for neural networks, nevertheless, it could be applied to virtually any supervised learning method, such as partially least squares or support vector machine. SA is a method which is used to evaluate the inputs and give them proper ranking depending on the changes of the outcoming prediction, given disturbances on each input. According to the functioning of SA, this method is normally used as a variable/feature selection method. In another paper, Cortez et al. also proposed another approach based on SA, where the author presented three new SA methods (data-based SA, Monte Carlo SA, cluster-based SA) and one input importance measure (Average Absolute Deviation) [2].

In addition to that, Goldstein et al. [35] proposed a method called Individual Conditional Expectation (ICE) plots which visualizes the model estimated by any supervised learning algorithm. Partial dependence plots (PDP) normally visualize the average partial relationship between one or more features with the outcome, ICE has the goal of clarifying partial dependence plot by describing the functional relationship between the final outcome

with the feature for individual observations in graphs. To be exact, ICE points out the variation in the fitted value across the range of a covariate, showing where or to what extent the relationship could be varied. Moreover, ICE also offers visual test for additive structure in the data generating model, which could provide much more useful information in comparison to classical PDPs [35].

In general, it is often difficult to create meaningful visualizations based on just the inputs and outputs of an opaque model. Moreover, since they are visualization methods for post-hoc explainability, they must also be able to be applied to any machine learning model, regardless of their inner structure. Therefore, most of the time, these visualization methods do not stand alone but are used in combination with feature relevance techniques, which helps provide the end users with essential information.

5.4. Transparent machine learning models

As mentioned above, interpretability has the goal of making sense of transparent models. In this section, different transparent machine learning models will be presented and analyzed, according to the three aspects defined by Lipton [50].

5.4.1. Linear/Logistic Regression (LR)

Linear/Logistic regression has the goal of forming the relationship between two variables by fitting a linear equation to observed data. A linear regression line has the equation of:

$$\mathbf{Y = a + bX}$$

X and Y are two variables of the equation, while X is the explanatory variable, Y is the dependent variable. A relationship between X and Y must be determined before trying to fit the equation into observed data. It does not necessarily mean that a causal relation has to be existed between the two variables, but they must be associated in a significant way. According to Arrieta et. al. [4]: “LR is a classification model to predict a dependent variable (category) that is dichotomous (binary).” This model, as mentioned above, attempts to assume the linear dependence between the predictors and the predicted variables, which makes a flexible fitting to the observed data a difficult task. For this exact reason falls this classification model under the category of transparent models. However, since there are different groups of users when it comes to AI models (see 1.2), depending on the purpose of who is trying to interpret the model, LR could be interpretable or explainable. In another word, even though LR fulfills all the requirements for a transparent model (algorithmic transparency, decomposability, and simulatability), there is still a possibility that post-hoc algorithms are required in order to explain the model, especially to non-expert users [4].

One of the most apparent advantage of LR is that it is totally transparent how a prediction is produced. The fact that many experts are using LR proves that this technique is approved for predictive modeling and doing inference [57].

On the other hand, LR could only be applied for linear relationships, any non-linear relationship has to be customized and fed to the model as an input feature. Due to the limited and simple learning possibility through linear relationships, the models are normally not outstanding in predictive performance. Moreover, complete separation could occur to logistic regression when a feature exists which would perfectly separate the two classes. At this point, it is impossible to train the logistic regression model. This is caused by the inability to converge that feature and therefore the optimal weight would be infinite [57].

5.4.2. Decision trees

Decision trees are another example of a transparent model as mentioned in the beginning of this paper. Decision trees follow a hierarchical structure in the process of decision-making which support regression and classification problems [60]. In these models, the data is split multiple times, through which different subsets of the dataset are generated. There are terminal or leaf nodes, which are the final subsets, furthermore, there are intermediate subsets which will be called internal nodes or split nodes [57]. The average outcome of the training data on one node is used to predict the outcome of that specific node. Interpreting a decision tree is a simple matter, one starts from the root node then making their way through each level of the tree. The predicted outcome is shown when the user arrives at the leaf node or terminal [57]. Generally, decision trees are simulatable models, however, according to their characteristics, the model could be decomposable or algorithmically transparent. It is important to mention that even though a decision tree has the capability of fulfilling every aspect of a transparent model, their individual properties could shift them more towards algorithmically transparent models. Simulatability by decision trees depends on whether the model is simple enough to be managed by human user, which means the trees could not be too complex, its size must be relative small, and it is uncomplicated to comprehend the amount as well as the meaning of its features. On the other hand, if a decision tree's size has exceeded the level of human understanding, the model turns into decomposable. In case the model continues to increase in size and feature relations become too complex, it becomes algorithmically transparent and the previous characteristics are no longer valid [4].

Decision trees with their customized transparency offer themselves as the perfect tool to support decision making tasks. Their structure is most suitable for grasping the relationship between different features in the data. It is normally easier to understand the data since they are divided in different groups. Moreover, since decision trees are basically a bundle of “what-if” sequences in a higher computer language, they offer a contrastive characteristic, one can always compare the prediction between different leaf nodes [57].

However, despite all the advantages mentioned above, Molnar [57] also mentioned some of their weaknesses. Firstly, it is not possible for decision trees to work with linear relationship since the relationship between input and output must be evaluated by the split. This requires a function on real numbers, which makes the whole algorithm inefficient. Furthermore, decision trees are quite unstable, since every split is dependent on the parent split, minor changes in the training dataset or a different parent split feature could result in a complete change in the tree structure itself. When the structure could easily change, the model loses its confidence.

5.4.3. K-Nearest Neighbors (kNN)

kNN is another classification method which is fundamental and is a suitable choice when there is little or no prior knowledge about how the data is distributed. Depending on the value of k, the class of a test sample will be decided by voting the existing classes in the data. A kNN algorithm follows basically three steps: calculate distance, find closest neighbors and vote for labels [60].

In the context of regression problems, an aggregation of the target value will be calculated in association with the nearest neighbors in order to find the class for this value instead of voting for labels like in normal context [4]. Since kNN produces predictions according to the distance or similarity between examples, these features could be customized depending on which problem is being analyzed. On the other hand, it is simple to explain kNN’s predictions since it is not so much different from human predictions based on past events. kNN is a lazy algorithm, which means no training data points will be used for the purpose of generalization. Since there is no generalization, all or most of the training data is necessary for the testing phase [14]. Like any other method, kNN has its own strengths and weaknesses.

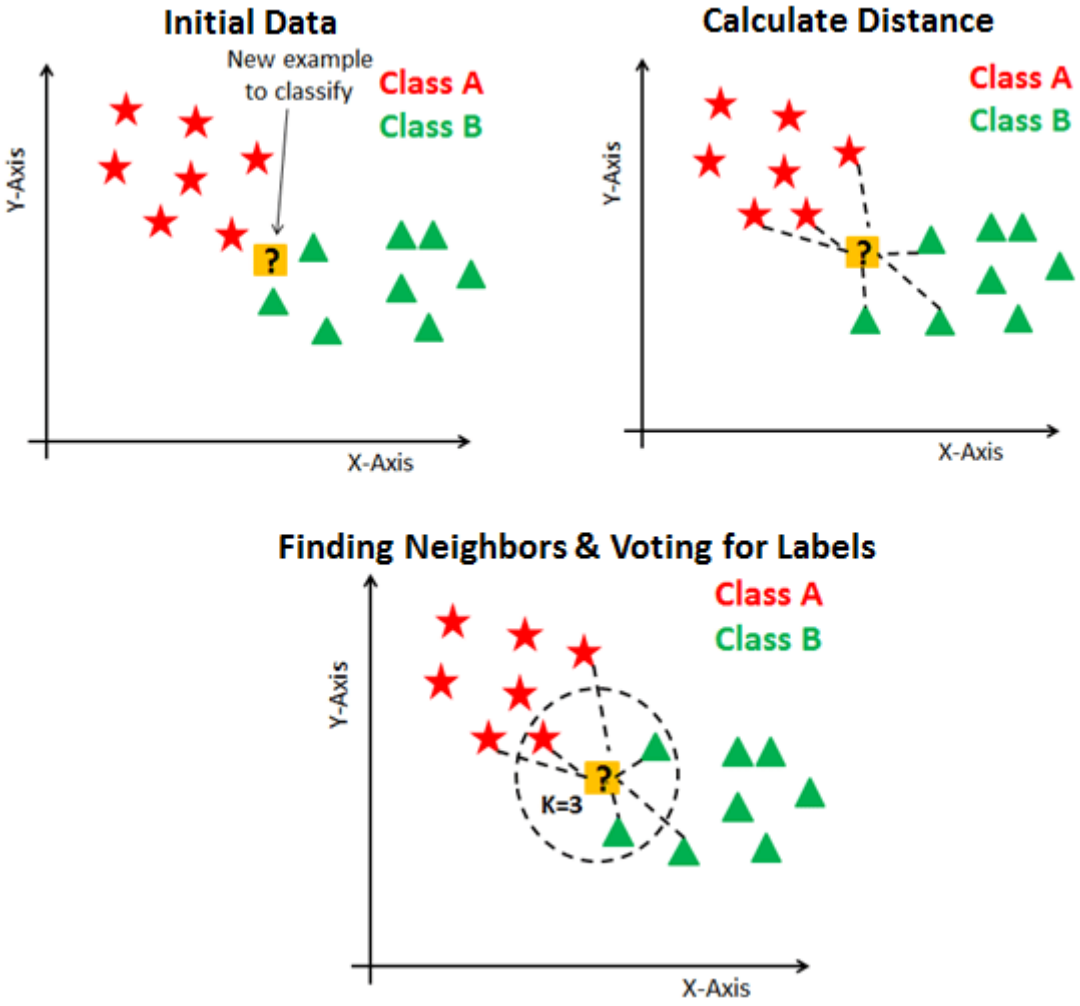

Figure 12: The three basic steps of kNN algorithm (Navlani 2018)

First of all, kNN ist very intuitive and simple, it is not only easy to comprehend the logic behind the predictions of a kNN algorithm, but it is also pretty simple to implement. As mentioned above, kNN combs through the whole training data points to search for the nearest neighbors in order to classify a value. kNN is a non-parametric algorithm, unlike linear regression, there are many assumptions to be met by data before implementing, kNN requires no assumption to be implemented. In addition to that, there is no training step in this method since it does not specifically build any model but only marks the new data entry based on learned data in the past. While most of classifier methods are only suitable for binary problems and require extra steps to adjust to multi-class, kNN can be used effortless to any multi-class problem. Another advantage of kNN has already been mentioned in the paragraph above, this algorithm could be applied to both classification and regression problems [14].

Aside from all the advantages, kNN contains some certain disadvantages. On one hand, kNN is a slow algorithm, even though it is simple to implement this algorithm, the bigger the dataset becomes, the slower and less efficient the model will be. On the other hand, it is difficult for this method to produce predictions when the number of variables grows too big. kNN also needs homogeneous features, all features are required to have the same scale, which means the distance applied to feature 1 must have the same meaning for feature 2. Additionally, kNN does not work well with imbalanced dataset, when one class is more dominant than the other, the model tends to prefer the class with majority of the features. This could lead to incorrect classification for the less common class. Finally, kNN cannot deal with the problem of missing value.

One important thing about kNN is that the features, the number of neighbors and the distance function used to measure the similarity between data instances are crucial to the method's class of transparency. When the features and/or the distance functions are too complex, kNN's interpretability is restricted to only the transparency of its algorithmic operations [4].

5.4.4. Rule-based learning models

Rule-based learning models generate rules to identify the characteristics of the data that the models are going to learn from. The goal of rule-based learning methods is to figure out the regularities in data which could be represented in a simple if-then rule [31]. The rules could be one single decision rule or a combination of several different rules which help with the process of predicting the output [57]. Fuzzy rule-based system is one category which belongs to this model family. Fuzzy rule-based systems provide a more flexible approach to tackle to problems. Basically, fuzzy rule-based systems are rule-based systems which contain fuzzy sets and fuzzy logic. These sets and logic provide support in representing different forms of knowledge about the problem that needs to be analyzed, as well as establishing the relationship and interactions between variables that already exist in the model [53]. One advantage that fuzzy rule-based system comes from the fact that fuzzy statements are being used as the main components of the decision rules, which creates a chance to identify possible uncertainty of the represented knowledge and eventually also handle any of the confusion that might occur.

A decision rule is considered to be useful if it fulfills two main criteria: coverage (support) and specificity (accuracy or confidence). Coverage of a rule refers to the percentage of the total variables to which a rule is going to apply. For example, if 5000 variables are being examined and only 100 out of 5000 fulfill the requirements of a rule for the output, then the coverage of that decision rule is 2%. On the other hand, the specificity of a rule refers to

the accuracy of the rule while predicting the correct class for the variables to which the rule applies. For example, out of the 100 variables which fulfill the rule above, only 80 of them have the same output as predicted while the rest 20 provide different prediction, in this case the accuracy of the rule is 80%. In general, the more features are being added to the rule, the higher the accuracy will be; however, the amount of coverage will be reduced accordingly [57].

As mentioned above, the rule could be singular or a combination of different rules. Molnar [57] stated that there are two main things that need to be taken into consideration while combining rules: decision lists and decision sets. A decision list offers a structure for decision rules. If the first rule applies to a variable, the output will be based on the prediction of the first rule. If this is not the case, the variable will be tested for the next rule until it fulfills one of the rules in the list. By using a decision list, the problem of overlapping rules will be avoided since the prediction only depends on the first rule that applies. Meanwhile, a decision set offers democracy amongst the rules, except some rules might have a higher influence on the process of decision-making. The rules in a set could be related to each other in a way that one excludes or precludes the other, or there might exist a solution in case conflicts arise, such as a majority voting. The rules will then be examined according to their accuracies or other quality criteria and a decision will be made as to which rule has the most influence and meaningful to the prediction. Nonetheless, while applying various rules, it is expected that the degree of interpretability of the model will decrease.

Like any other transparent models, rule-based models also have their advantages and disadvantages. Rule-based models are normally highly transparent and expressive since these models inherit the advantages of if-then rules. If-then rules are extremely interpretable, however, this case is only true while the number of if-statements are limited, or the rules are coordinated in a decision list or in a non-overlapping decision set [57]. By applying if-then rules, the process of reaching a prediction is fast, since the only statements which need to be checked are binary. In addition to that, readability, maintainability and most important the possibility to transfer domain knowledge directly into rules is another advantage of this model [87]. Since in rule-based models, it only matters if a statement is fulfilled or not, they are very robust against outliers. Finally, if-then rules only select features which are relevant to the model since there are not many features included in the model [57].

Aside from the advantages, despite its simplicity, rule-based models have a few restrictions. Firstly, if-then rules mostly completely ignore regression and only focus on classification. Moreover, even in classification problems, despite having the possibility to divide a continuous target into smaller intervals, it is unavoidable that information will be lost during the process. On the other hand, most of the time the variables in rule-based

models must be explicit and direct, which means numbers must be categorized before they could be used in the model. Furthermore, rule-based models are not suitable for presenting the linear relationships between variables and predictions. This is a feature similar to decision trees since in both models, only step-like prediction functions are being applied and the inputs have to be absolute [57].

5.4.5. Generalized additive models (GAM)

GAM is an extension of generalized linear models which use smooth functions of predictor variables for the integration of nonlinear forms of the predictors [19]. Traditionally, likelihood-based regression models assume a linear form for the covariates. Meanwhile, GAM takes out the linear part of a model and replaces it with a sum of smooth functions. A number for unspecified smooth functions is going to be aggregated using a scatterplot smoother and through local scoring algorithm [40]. Sequentially, the value of the variable to be predicted will be given depending on that number. The class of generalized linear models contains many of likelihood-based regression model, in GAM, the linear predictor will be replaced by the additive predictor, therefore it is called generalized additive models. GAM has proven in different cases to be useful in detecting nonlinear covariate effects.

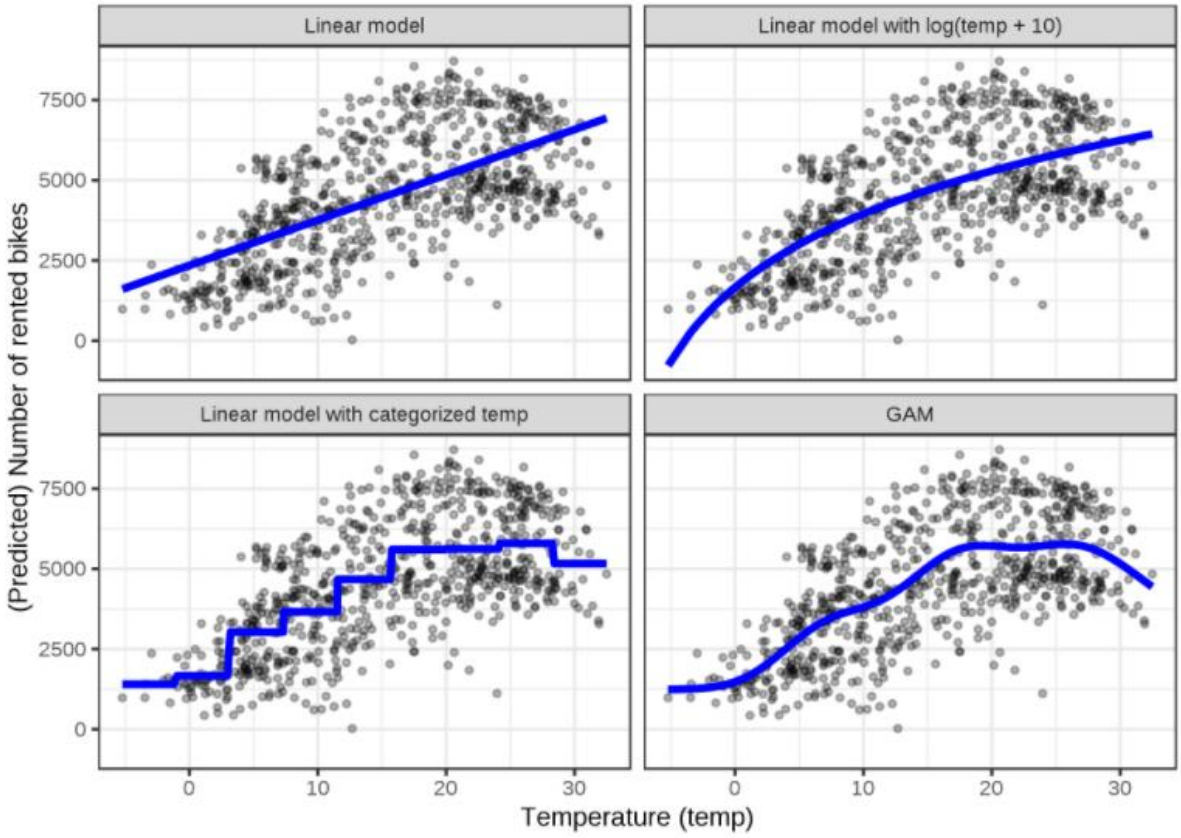

Figure 13: Predicting the number of rented bicycles using only temparature feature with different models (Molnar 2020)

According to Figure 13, a linear model is completely not suitable for this type of data. Transforming the model with a logarithm (top right) or categorization (bottom left), which proves to be not so effective. With generalized additive model (bottom right), the curve for temperature is automatically smoothened and it fits a lot better to the data [57].

GAM is not a new method in the field of machine learning, this transparent model has been around for over 30 years, which makes it an advantage for this model. Experts and practitioners are more comfortable using GAM as a technique for modeling since they already have a lot of experience with this model. Comparing to other transparent model, GAM is comprehensible enough to be applied in different fields like finance, environmental studies, geology, healthcare, biology, and energy [4]. These applications all have one thing in common, which is understandability. GAMs offer these studies a chance to examine the underlying relationships that cause the cases that need to be inspected, where accuracy is not the focus but to be able to understand the problem behind as well as the relationships underneath the related variables [4]. Aside from using the model to make prediction, drawing conclusions about the data is also another possibility for model usage, considering that none of the model assumptions are violated. As mentioned above, GAMs have been around for over 30 years, therefore a lot of statistical software already implemented interfaces to make it easier to apply GAMs.

Since GAMs is basically an extension of linear models, any modification to a model most likely would make the model less interpretable. Applying GAMs or various smooth functions makes it impossible to summarized nonlinear feature effects by a single number [57]. On another hand, GAMs generate predictions based on assumptions about the data, in case these assumptions are violated, the interpretation of weights is no longer eligible.

In conclusion, despite the transparency of those models above, which makes them interpretable, the border between interpretable and explainable are still vague. The models and techniques mentioned above are just some typical examples of interpretable models, the field of interpretability is still being explored and researched. Depending on the purpose of understandability, a transparent model might require extra steps (e.g. post-hoc techniques) to be able to satisfy the users' needs. Therefore, an interpretable model could also become explainable if the purpose of understandability requires it. In the next section of this paper, shallow machine learning models will be presented which are transparent but still requires post-hoc explainability to make the predictions comprehensible.

5.5. Post-hoc explainability for shallow machine learning models

Shallow machine learning models are exceptionally good in covering many aspects of supervised machine learning. The most popular models in term of shallow machine learning which requires post-hoc techniques for prediction explanations are tree ensembles and support vector machine (SVM) since they have proven to excel in performing predictive tasks.

5.5.1. Tree ensembles, random forests, and multiple classifier systems

Traditional decision trees possess poor generalization properties which makes the model less attractive for experts to apply them when a balance between predictive performance is an essential feature. Tree ensembles offer the possibility to improve this trait of decision trees. A group of decision trees are called random forest (Figure 14), which could be considered a special case of bagging. This method serves the purpose of constructing a sample in a large amount of data as quickly as possible. The predictions performed by trees which result from learning from different subsets of training data will be aggregated. In comparison to other ML models, tree ensembles could be considered to generate most accurate outcomes. As mentioned above, single decision trees have poor generalization capability which normally leads to overfitting, meanwhile, tree ensembles get around this problem by merging different trees together to acquire an accumulated prediction/regression. While this type of model has proven to be effective against overfitting, the combination of trees, on the other hand, makes the comprehensibility of the overall model more complex than a normal decision tree and therefore the model loses its transparency. As a result, tree ensembles require post-hoc explainability techniques to explain their predicted outcome, in particular explanation by simplification and feature relevance [4].

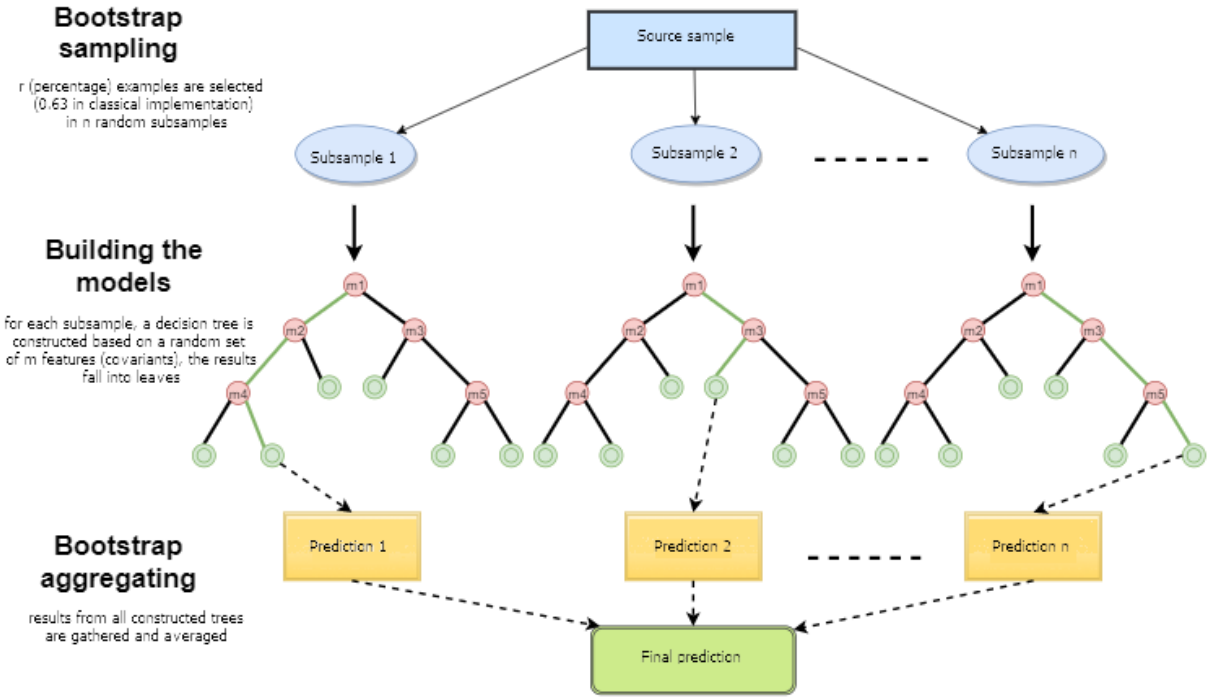

Figure 14: Diagrammatic representation of a random forest (Aldrich 2020)

CMM is an example for explanation by simplification as a meta-learner which maintains most of the accuracy while making tree ensembles simpler to understand by generating one single comprehensible model [24]. In his paper, Domingos also mentioned the process of applying a value of $m \gg 1$ to m different training sets, where two different training sets are consisted of the same features but with different weights or to m different versions of the classification learner itself, which is limited to m different learners. Afterwards, the results of m models will be combined in some form as one single model.

Another framework was proposed by Deng [22] called inTrees (interpretable trees) where rules from tree ensembles are extracted, evaluated, unrelated or unnecessary value pairs of a rule will be eliminated, and a set of relevant rules will be selected (Figure 15). Afterwards, a rule-based learner, for example the simplified tree ensemble learner (STEL), could also be created and used for summarizing rules as well as predicting new data in the future.

Hara et. al. [39] also proposed a post-processing approach to make tree ensembles explainable. The idea was to let tree ensembles functions normally where a number of regions are generated, subsequently, the behavior of tree ensembles will be imitated by a simple model with a fixed number of regions. The learned tree ensemble model is responsible for prediction and the mimic model for explanation.

Aside from explanation by simplification, tree ensembles could also be made explainable through feature relevance. The first paper to analyze the variable importance in random

forest was written by Breiman [13] where he proposed a method based on measuring MDA (Mean Decrease Accuracy) or MIE (Mean Increase Error) of the forest when a certain variable is randomly changed in the samples (out-of-bag samples) which are not included in the subsets of data (bootstraps samples). Other authors, such as Auret et. al. [5], also contributed to Breiman's method, which represents how variable importance could reflect the relationships underneath the complex structure of a random forest.

Meanwhile, Tolomei et. al. [84] proposed an algorithm which functions on top of any ensembles of trees. This method adjust the input features so that the output predictions of a machine learning model will be affected, the algorithm takes advantage of the internal functioning of a model to create suggestions, which could turn a variable from one class to another. In the research paper, the authors first designed a random forest classifier to separate online advertisement between low (bad quality) and high (good quality), sequentially, an algorithm will be applied in order to make comprehensible suggestions to optimize a low quality advertisement and convert it into a high-quality one. The complete process will be evaluated in the end on a subset of the database of Yahoo Gemini and presented visually [84].

In conclusion, amongst all post-hoc techniques for shallow machine learning models, for random forest or tree ensembles, explanation by simplification and feature relevance stay the most popular and favorite techniques.

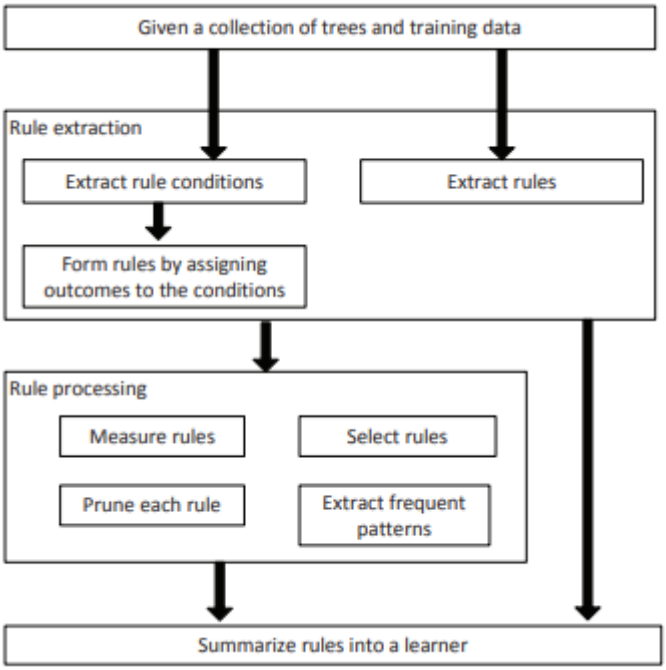

Figure 15: Illustration of the inTrees framework (Deng 2014)

5.5.2. Support Vector Machine (SVM)

Support Vector Machine is another shallow ML model which has been a research topic since 1979 by Vapnik. The main objective of SVM is to find a hyperplane or set of hyperplanes which explicitly classifies the data points in a n-dimensional space in two classes, with n equals the number of features [12]. SVM could be applied for both regression and classification problems. Basically, there are many possibilities for hyperplanes to separate the two classes of data points. However, an optimal hyperplane is one that has the maximum distance (also known as functional margin) between data points of both classes [4]. Maximum margin also guarantees that further classification of data points will be operated with more confidence. SVM is a much more obscure system than tree ensembles, therefore, post-hoc techniques are also required by experts to comprehend the internal mathematical algorithms. The most frequently used post-hoc methods for explainability are explanation by simplification, local explanations, visualizations, and explanations by example [4].

Arrieta et. al. [4] mentioned in his paper that there are four classes of explanation are created for explanation by simplification method depending on how deep the methods go into the inner structure of SVM. The first class attempts to generate a rule-based model solely based on support vectors of a trained model. Barakat et. al. [8] proposed in his paper a rule extraction technique from SVM's. The technique is called SQRex-SVM, where a modified sequential covering algorithm is applied to extract rules directly from the support vectors of a trained SVM. This technique has proven to provide better generalization performance and the rule sets generated are smaller and more understandable compared to other SVM rule extraction techniques [8]. Following this path, Barakat published another paper proposing another approach called eclectic rule extraction, which also extract rules directly from support vectors of a trained SVM [9]. Barakat explained that his method operates in three steps: learning stage – rule generation – evaluating the quality of the extracted rules. In the first step, the SVM model will be trained using labeled patterns in order to generate a classifier with justifiable accuracy, precision and recall. The second stage serves the purpose of represent the information learned by the model in a way that the users could understand. This stage requires two extra steps, the first step tries to take advantage of the knowledge presented by support vectors and parameters related to them. The second step attempts to describe the acquired knowledge in a comprehensible form. The final stage of this technique deals with evaluating the quality of the extracted rules by using a second dataset.

The second class of explanation by simplification is proposed by Fu et. al. [30] where the authors created the hyper-rectangles based on the cross points between each support vector and the hyperplane. A tuning phase is responsible for adjusting the hyper-rectangle, so that out-class data points will be eliminated. Redundant rules are eventually combined to create a compact set of rules [30].

The third class involves using the actual training data as a part of the components for generating the rules. SVM + Prototypes method is also a rule-based method, which was mentioned in different research papers [64,62,71]. This method makes it possible for SVM to generate explanations using a clustering algorithm to group prototype vectors for each class. This method makes use of the information extracted from the support vectors [62]. These support vectors are responsible for identifying the boundaries of regions defined in the input space, where they are merged with prototype vectors using geometric methods in order to transform these regions into if-then rules [63].

The final class takes advantage of an existing solution to provide explanation to SVM decisions [4]. This method is known as growing support vector classifier (GSVC), which builds rules as a linear combination of input variables [59]. These rules define the space in Voronoi sections, which are decided by the extracted prototypes of input variables [4].

Visualization is another technique which is also being used to explain SVM models when concrete applications are involved. One method offers the chance to visualize the extracted information of the kernel matrix in a trained SVM and with that a possibility to explicitly explain the instances of the support vector regression model [101]. In another research paper, the authors attempted to explain linear SVM models of large-scale data sets with heat map. The idea was to assign different colors to each atom and bond of a compound depending on its importance for activity based on the weights of a trained linear SVM model to support early drug discovery stage [75].

Wenzel et. al. [89] proposed an interesting approach by merging nonlinear SVMs with Bayesian systems to create a fast, scalable, and reliable approximate inference method for large amount of data. Bayesian methods have always been popular for the ability to absorb prior information and produce accurate probability calculations. By combining nonlinear SVMs with Bayesian analysis, it is possible to extract the best traits of both methods: geometric interpretation, robust against outliers, high accuracy, theoretical error guarantees from the frequentist formulation of the SVM; flexible feature modeling, automatic hyperparameter tuning, and predictive uncertainty quantification [89].

5.6. Explainability in deep learning systems

5.6.1. Multi-layer Neural Networks (Multi-layer Perceptrons)

In 1958, Rosenblatt [76] proposed the classical perceptron model, which was then developed further by Minsky et. al. [56] and became the perceptron model. The inputs in perceptron model introduced by Minsky have numerical weights, which help determine which input has more importance on the output, which was not introduced in earlier perceptron model before.

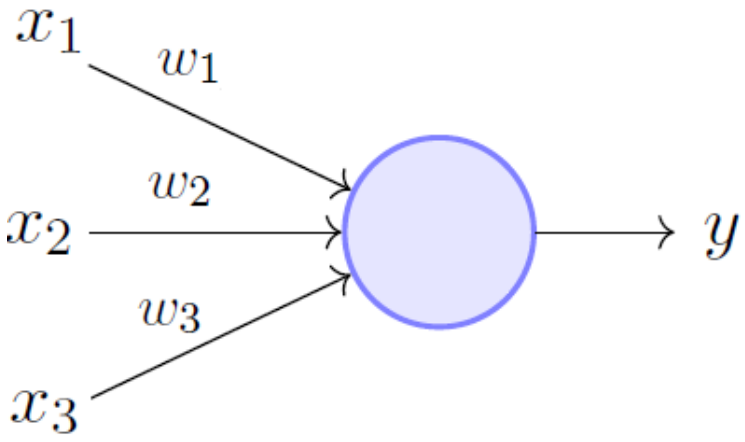

Figure 16: The perceptron model (Minsky et. al. 1969)

However, one perceptron does not possess the ability to solve non-linear classification problems, hence, a generic model is required which has the ability adapt to some training data. For these reasons, multi-layer neural network was invented, which was also known as multi-layer perceptron. This type of neural networks has one input layer, one or more hidden layers and one output layer.

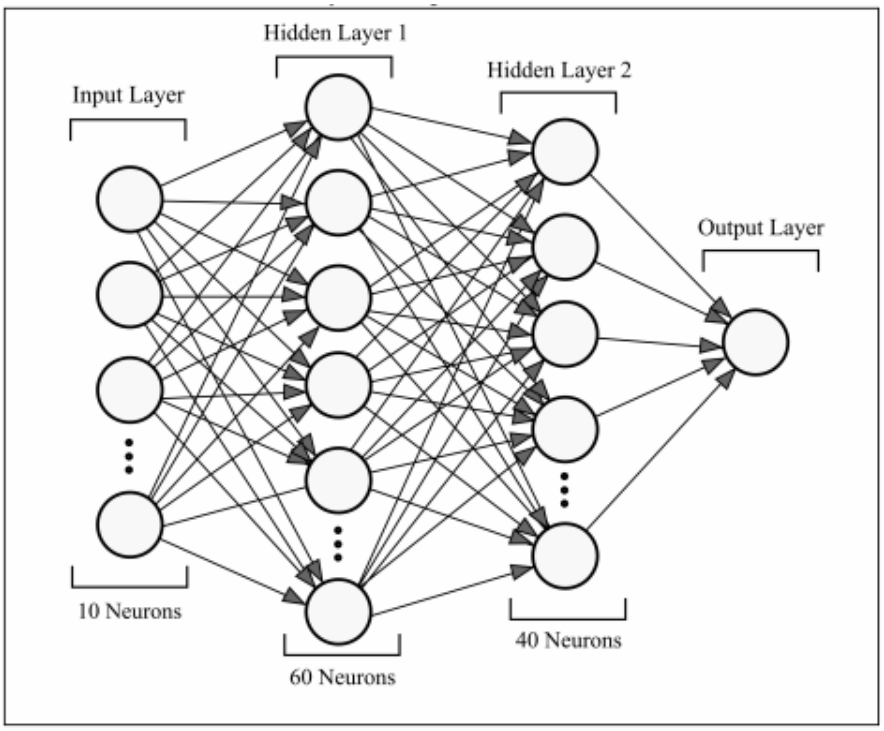

Figure 17: Multi-layer perceptron architecture (Oliveira 2014)

Multi-layer neural networks are also known as feed-forward networks, which means every neuron in the same layer is connected to all neurons in the next layer, but not to each other, and the information flows from input layer to output layer [66].

Despite their excellent performance in figuring out the complex relations among variables, multi-layer neural networks have always been treated as black-box models since it is almost impossible to know what is happening inside the hidden layers of a neural network. Nevertheless, as technology advances, a model is not only valuable for its accuracy, but also for its explainability. Researchers have been working on methods to explain deep learning models such as multi-layer neural networks. For these complex models, post-hoc techniques such as explanation by simplification, feature relevance, text explanations, local explanations and model visualizations are normally applied to produce comprehensible reasoning for their outputs.

Arrieta [4] mentioned that several approaches with explanation by simplification were proposed for neural networks with one hidden layer, however, there are not many research papers for those with multiple hidden layers. For multiple hidden layers neural networks, it is essential for users to understand the process of decision-making and the outcome predictions, especially in fields such as medicine, autonomous driving, or financial markets.

Zilke et. al. proposed a solution for this problem, in which the behavior of the neural networks is portrayed through decision rules [100]. According to the authors, the goal of deepRED is simply to access hidden features of DNNs and take advantage of their deep structure to enhance the ability to produce desired results of rule extraction and induction process.

In order to understand how deepRED works, one must get to know the process of Continuous/discrete Rule Extractor vis Decision tree (CRED), which was presented by Sato and Tsukimoto [78]. CRED can extract rules that has both continuous and discrete variables in data. When decision trees deal with classification problems, the outcome data is discrete or is categorized (approval of credit, a person died or survived), meanwhile, when decision trees deal with regression problem, the outcome data is continuous (house prices, age of a person). In the first step, CRED will identify the question for a neural network, depending on the network type (continuous or discrete). In the next step, a decision tree (hidden-output tree) is created based on activation pattern of hidden units as attribute data, and the discretized pattern from first step as class data. Exclusive production rules (intermediate rules are then extracted from decision tree and the network will be broken down into constitutive functions. The next step creates decision trees (input-hidden trees) where each of the tree correlates to a function generated in the previous step. The attribute data here is the input variables of given data and the discretized pattern from the second step is the class variable. From each input-hidden tree, production rules (input-rules) are created, which make the function of the hidden unit more understandable, each rule will then be simplified, and redundant rules will be eliminated. In the fourth step, the intermediate rules are replaced by the input rules, creating the total rules, which represent the relationship between input pattern and target query in the first step. Once again, redundant rules are excluded after each total rule is simplified in the last step of CRED.

The reason why the process of CRED must be presented is that DeepRED basically extends CRED algorithm to multiple hidden layers. Since CRED only deals with neural networks with one hidden layer, DeepRED offers the possibility to solve the explainability problem of neural networks with unspecified hidden layers, in which this method applies CRED layer by layer [100].

Another approach called interpretable mimic learning, which also attempts to explain multi-layer neural networks was mentioned in the paper by Che et. al. [17]. Similar to the “Distill-and-compare” method from Tan et. al. mentioned in section 5.3.1, this method firstly filters the knowledge from the teacher model (a complex, slow, but accurate model) and pass it on to the student model (a much smaller, faster, but still accurate model). In addition to that, the authors made use of gradient boosting trees (GBT), an ensemble of decision trees, as a tool to make deep learning models comprehensible.

There are also other papers expanding existing research on explaining multi-layer neural networks with simplification, however, the problem becomes more complicated as the number of layers increases. Therefore, experts have shifted their direction towards feature relevance methods. Montavon et. al. [58] proposed deep Taylor as a feature relevance method to tackle deep neural networks. This is a decomposition method which is based on divide-and-conquer paradigm and breaks down the property that the function learned into a set of subfunctions. Each neuron acts as an object that could be broken down and expanded then aggregate. Back propagation is applied to these decompositions through the network, which creates deep Taylor decomposition [4]. Another feature relevance method which follows the same direction with the research of Montavon is deepLIFT (Deep Learning Important FeaTures) [80]. This method attempts to make comparison between each neuron's activation and its 'reference activation' then allocates their contribution scores based on the difference. Since this method differentiates between positive and negative contributions, it offers the possibility to identify dependencies which are normally overlooked by other methods.

5.6.2. Convolutional neural networks (CNN)

Convolutional neural networks could be considered one the most impressive forms of Artificial Neural Networks (ANN). CNNs have proven to achieve excellent performance in all fundamental computer vision tasks, such as image classification, object detection or instance segmentation. This type of model is similar to traditional ANNs since they also consist of neurons that self-optimize through learning, however, CNNs are primarily applied when it comes to pattern recognition within images [67].

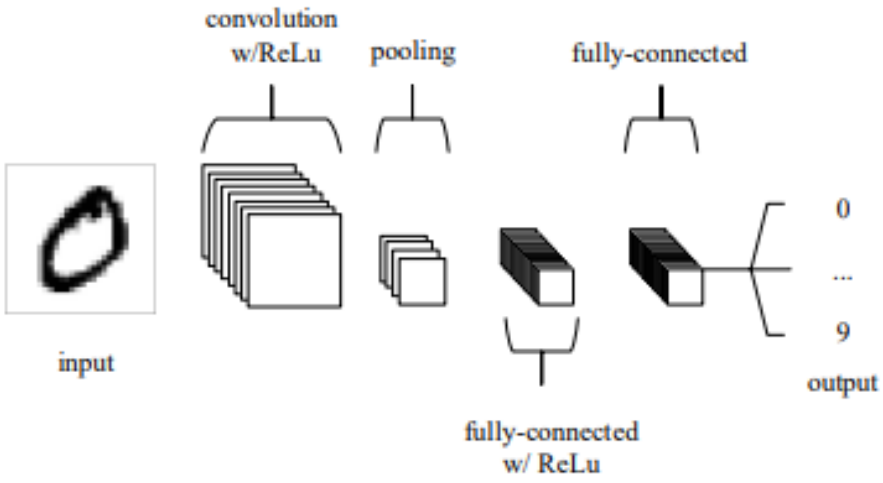

Figure 18: A simple CNN architecture (O'Shea et. al. 2015)

As can be seen on Figure 18, a typical CNN model possesses three layers: convolutional, pooling and fully-connected. Convolutional layers and pooling layers are responsible for learning higher level features automatically, meanwhile fully-connected layers are placed at the end of the sequence to map the outputs into scores. There are two different approaches while attempting to comprehend CNN models, either to understand the whole process of decision-making of the model by mapping the output back in the input space to identify which input is considered crucial to the output prediction, or to understand the internal structure of the model, how each layer of the model perceives the external world, regardless of the inputs.

With the purpose of figuring out the reasoning behind CNNs' predictions, Zeiler et. al. [96] proposed a framework called Deconvolutional Networks (Deconvnet), which was later further developed by the same authors themselves. The original deconvnet framework allows unsupervised construction of hierarchical image representations, which is suitable not only for low-level tasks, but also offers features for object recognition. In their latter research paper, Zeiler et. al. [97] expanded the original deconvnet framework and focused its purpose on mid and high-level feature learning, a model which could rebuild the maximum activations when fed with a feature map from a selected layer [4].

Global average pooling is also a method which was originally designed to avoid overfitting, however Zhou et. al. [99] discovered that this method, with a little change, has the capacity to determine the discriminative image regions without difficulty. The authors presented a technique called Class Activation Mapping (CAM) for CNN models with global average pooling. This technique allows the users to visually see the predicted class scores on any given image and emphasizes the regions that the CNNs considered important to the image classification, therefore provide users more or less a comprehensible explanation of the predicted outcome. In addition to that, the authors also confirmed in later research that convolutional layers could be replaced by max-pooling layers without losing the model's accuracy. Figure 19 shows an example while applying CAM on a given image, the ground truth is labeled as dome. According to observation, class 'dome' activated the upper round regions in the image, meanwhile class 'palace' activated the lower flat part of building

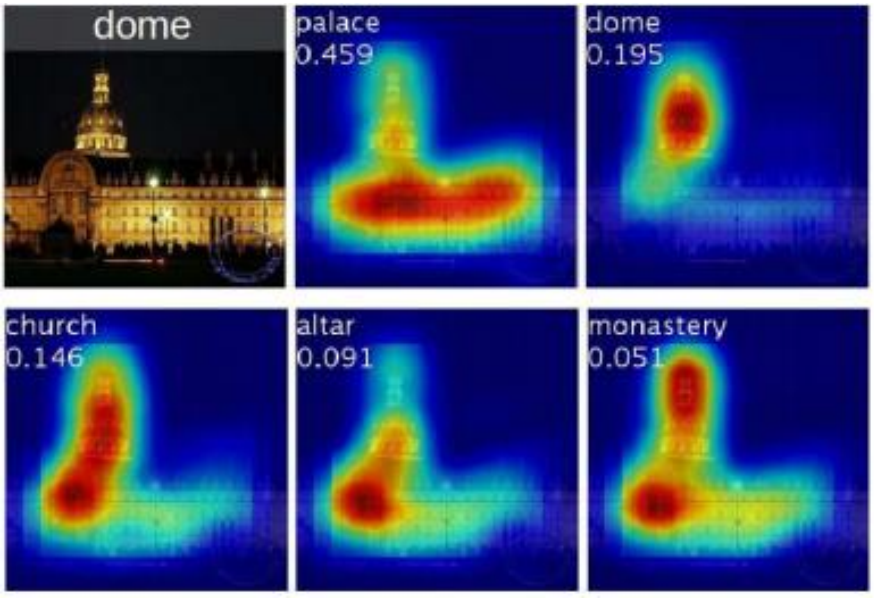

Figure 19: Examples of the CAMs generated from the top 5 predicted categories for the given image (Zhou et. al. 2016)

Zhang et. al. [98] introduced in their paper a method for interpretable convolutional neural networks, however, since this method is attempting to tackle the logic behind the model's prediction and not to understand the internal structure of the model, the authors are actually implying the explainability in convolutional neural networks and not interpretability. By applying a simple but effective loss to each filter in a specific conv-layer to force a layer to represent an object part, this method can point out how the object parts are distributed, which the CNN model considers to be relevant for object classification.

Another method contributing to CNN explainability is called pixel-wise explanation, in which the authors tried to break down the image in to single pixel to determine how much weight each pixel has on the outcome prediction, whether positive or negative [6]. The authors also created a heatmap to visualize the pixels' contribution to the image classification.

Aside from feature relevance and visual explanation methods, text explanations of the visual representation of the image are also taken into consideration while explaining CNNs. According to the work by Xu et. al. [94], the authors introduced an attention-based model which has the ability to automatically generate text description of images. The method proposed in this paper was a combination of a CNN feature extractor and an RNN attention model, which has the goal to identify where and what the attention focused on and generate a caption based on whether the attention is useful after validating. A two-level attention model based on the work of Xu et. al. was presented in the paper by Xiao et. al. [92] in which the authors designed the technique based on simple intuition. Firstly, the system should be able to identify what kind of object that is and then look for the important features that distinguish that object from others.

There are already some approaches proposed by different authors when the users do not concern about the input, but the functioning of the layers in the model. One method uses an approximated inverse to understand deep image representation [54], where the aim is to figure out to which extent it is possible to reconstruct the image itself, given an encoding of an image. The authors therefore implemented a framework that rebuilds an image from the CNN internal representations and demonstrates that several layers in CNNs maintain photographically accurate information about the image. Similarly, Deep Generator Network (DGN) is also a technique which aims to create the most representative image for a given output neuron in a CNN [61]. The authors focused on understanding the inner workings of Deep Neural Networks (DNNs) by searching for the favorable inputs for each of the neurons.

Olah et. al. [65] presented a combination of different methods to acquire more information about the network. As an example, the authors combined feature visualization (what is a neuron looking for?) with attribution (how does it affect the output?) to analyze how the network decides between labels. The interface proposed by Olah et. al. is a mixture of blocks which follows a structure based on what the interface is trying to achieve, whether it is to focus on showing what the network detects, emphasize how the network develops its understanding, or concentrating on keeping things human-scale. One could consider this interface as a combination of elements, in which each element represents a particular type of content (e.g., the amount a neuron fires or which classes a spatial position most contributes to) using a distinctive style of presentation (e.g., information visualization, feature visualization). Moreover, the elements belong to layers (input, hidden, output) and atoms (e.g., a neuron, channel, spatial or neuron group).

5.6.3. Recurrent Neural Networks (RNNs)

Similar to CNNs, RNNs are also getting more attention due to their performance in natural language processing and time series analysis, since RNNs' structure is suitable for data that is sequential in nature [18]. RNNs can accept inputs as a series while adding additional layer of comprehension on top of the previous input, instead of receiving inputs altogether as a set. RNN architectures range from fully interconnected to partially connected, consisting of multilayer feed-forward networks with specific input and output layers [55]. A fully connected RNN does not possess a distinct input layers of nodes, furthermore, each node receives an input from other nodes.

While attempting to make RNN models comprehensible, experts have been doing research in two directions: either by understanding what the model has learned or by modifying the model's architectures to retrieve information about the decisions which were made.

In the first group, Arras et. al. [3] extended a technique called Layer-wise Relevance Propagation (LRP) to deliver resourceful insights in the form of input space relevance for understanding RNNs. The authors applied a specific propagation rule which works with multiplicative connections as they appear in recurrent network structures such as Long Short Term Memory (LSTM) units and Gated Recurrent Units (GRUs). Following the goal of keeping the architecture intact, Che et. al. [16] introduced a novel knowledge-distillation technique known as Interpretable Mimic Learning. This framework imitates the performance of deep learning models and uses Gradient Boosting Trees to learn their interpretable features while maintaining to generate robust prediction.

A research which belongs to the second group is Reverse Time AttentIoN (RETAIN), a method which achieves high accuracy while remaining interpretable and uses a two-level neural attention model to identify past patterns and important variables [18]. Also belong to this group, a combination of RNN and sequential iterative soft-thresholding algorithm (SISTA), also known as SISTA-RNN, which designs a sequence of correlated observations with a sequence of sparse latent vectors [91]. This approach produces a novel stacked RNN architecture with interpretable weights as the parameters of a principled statistical model. Lastly, by combining an RNN with a Hidden Markov Model (HMM), Krakovna and Doshi-Velez [48] have created a much simpler and more transparent model. In their paper, the authors experimented with different combinations of RNNs and HMMs to achieve both the interpretability of HMMs and the predictive power of RNNs.

5.7. Transparent and black-box models hybrid

As mentioned in the previous sections of this paper, deep learning models or black-box models are often too complex to be understood in comparison to transparent models. There are attempts to delve into the internal structures of these models, however, since the algorithms logic is mostly learned from data and is normally not presented in the source code, it is almost impossible to get behind what was really happening throughout the whole process. Instead of trying to make sense of the technical structures of neural network models, experts are now more focused on acquiring basic insights on the features that lead to the final decisions. Black-box models are less about interpretability and more about explainability, which means, a model should have the ability to present the relationships between inputs and outcomes, how much a particular input could affect the prediction, identify possible biases and recommend solutions for existing problems.

Wang [88] proposed an alternative solution for making black-box models understand for humans, which is called Hybrid Predictive Model. This method uses interpretable partial substitute to work on a subset of data, where an interpretable model is able to generate

predictions which are as good as the black-box model. The interpretable model offers explanations either at no cost of losing predictive performance or when the user does not mind trading accuracy for transparency, the model could identify the right subset of data at minimal cost of predictive performance. In another word, the author designed a framework where an instance input first goes through the interpretable model, when the model could not produce a prediction from that input, the black-box model will then be activated [88].

Another example of classical machine learning model combined with deep learning is Deep Nearest Neighbors (DkNN) proposed by Papernot et. al. [68]. DkNN is a combination of k-nearest neighbors algorithm and the representations of the data learned by each layer of the DNN. The method conducts a nearest neighbor search for each layer in the DNN to find training points for which the layer's out put is closest to the layer's output on the test input of interest [68]. On the other hand, the neighbors also constitute human-interpretable explanations of predictions, which supports the model with credibility.

III. Challenges and opportunities

The final part of this paper is focusing on the challenges and opportunities that XAI is facing, as well as an outlook in the future of how technology would affect mankind in general, especially in the field artificial intelligence.

1. Challenges of XAI

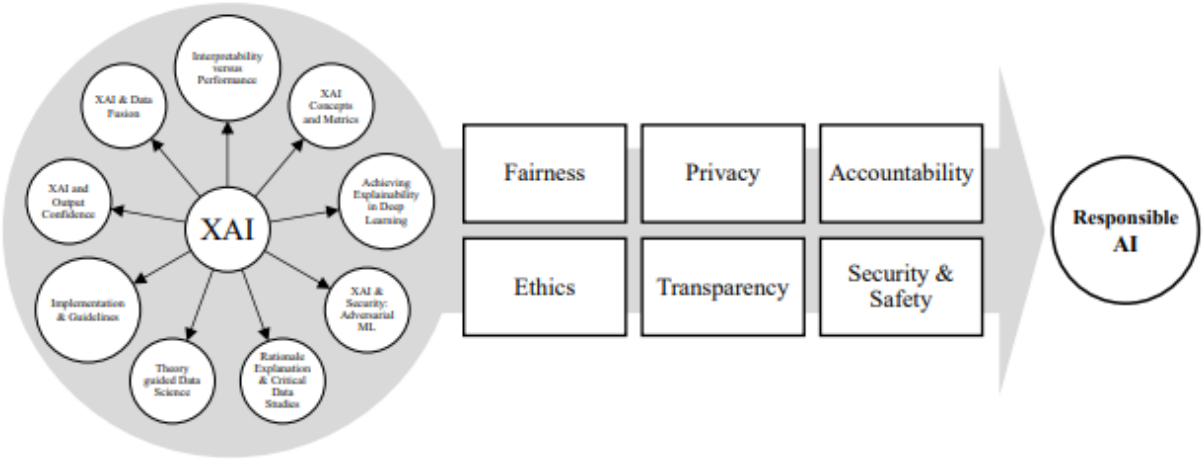

Figure 20: XAI challenges towards Responsible AI (Arrieta 2019)

Since technology is advancing at a rapid rate and mankind is getting more and more dependent on the development of it, XAI also gains more attention and a lot of researchers are investing time as well as effort to unbox this field. Creating comprehensible AI systems open a horizon for new opportunities and advances. However, as mentioned in different paragraphs of this paper, it is not always easy to implement a system that could

independently explain itself. There are many challenges an AI system must face while attempting to achieve explainability/interpretability (Figure 20).

One of the problems when it comes to machine learning is system bias. While feeding machine learning systems with a huge amount of data, it is essential to ensure that the system's decisions will not be affected by the deficiencies of the training data, model, or objective function. These deficiencies can cause unwanted biases, which could either be a biased view of the real world or an unbiased view of biased world, intentionally or unintentionally.

Another issue which needs to be dealt with is to maintain the fairness of the systems. In a sense, fairness is related to the problem of biases, one has to verify which condition or inputs actually contribute to the decisions once they are made. Every decision which is generated by an AI system must be made fairly depending on the data input provided to the learning algorithms and should not contain any discrimination against individuals or organizations in terms of race, gender, religion, sexual orientation, disability, ethnic, origin or any other personal matters [4]. The issue of fairness is extremely important when it comes to implementing an AI system, therefore, identifying biases in a system is crucial. Fortunately, there is a possibility to apply XAI techniques to detect biases, for example using SHAP to create counterfactual outputs which show the reasoning behind a model's decisions when being fed with protected or unprotected variables. By taking advantage of this information and determine the correlations between these features, one could disclose the reasoning behind unfair decisions or discrimination [4].

On the other hand, questions about transparency should also be raised in order to figure out to what extend should users have access to the explanation as well as how do they acquire those explanations. Moreover, can the users rely on the decisions of the AI systems without knowing the reasoning behind it? In another word, when the data cannot be seen or analyzed, is there a possibility to fathom the errors which are detected in the algorithm. As user, one must have the knowledge that he/she is interacting with an AI system or a person or what kind of information is being used by the AI system and for which reason or purpose. Depending on different groups of users, one possibility is to apply XAI techniques which have been mentioned in this paper to make the AI systems more comprehensible to a certain degree. Additionally, it is also essential to be able to debug incorrect output from a trained model.

Since AI systems are normally provided with huge amount of data, which contains sensitive information. Hence, it is required that one pays strict attention to the matter of privacy and security standards throughout the process in which the data is used to protect the system from any vulnerability [23]. The privacy of data always needs to be maintained, not only for

information, which is provided by users, but also information which is produced by the system about the users according to their activities. Measures to ensure privacy and security need to be developed and adapted to avoid potential threats, additionally, regulations and specific rules should also be implemented to keep everything under control.

2. Opportunities of XAI

The challenges that XAI is facing are at the same time offering a lot of opportunities for businesses to take advantage of.

First of all, XAI offers future innovations a chance to collaborate. Organizations and groups of researchers, experts, developers, and users are brought together to make sure that any advances in technologies will be serving the benefit of people and society. Any problems or concerns which might surface will be analyzed properly in different levels. This union also bring people together to work on achieving the opportunities and possibilities of the desire of conquering the computational science of intelligence.

As stated throughout this paper, the existence of AI systems that outperform humans in specific fields is already there and will continue to expand to a common matter. Since there might not be a comprehensible reasoning to the exceptional performance of AI systems, it is harder for these systems to be accepted by human. Therefore, the research on XAI topics needs to be carried on so that mankind could benefit from what these systems could offer. Moreover, when the logic behind an AI system could be understood, it is easier for the users to learn how to work with those AI systems when they outperform human in specific domains.

The implementation of XAI methods and techniques enhances the trustworthiness of the predictions of AI systems. The more reliable an automated system is, the more opportunities would be provided for further development. On the other hand, since it is very probable that mankind would be even more dependent on automated systems to make decisions in the future, it means one should be more explicit and structured about the principles or values that decide how the prediction will turn out.

IV. Outlook and conclusion

The first goal of this paper is to give a general introduction on the field of eXplainable Artificial Intelligence (XAI). XAI has become more and more important in the technology industry and has awaken a lot of attention from researchers, experts and even end-users. Not everyone agrees to see XAI as necessary and there are contradicting opinions on this. Therefore, arguments are presented to clarify the importance of XAI and the reason why it is absolutely necessary for this topic to be further researched. Different user groups also contribute to the extent of explainability/interpretability since the purpose of needing XAI decides how much or in which way a system should be comprehended.

System reliability and vulnerabilities which could occur were also mentioned in this paper as a crucial criterion while developing an AI system. By evaluating the performances of the systems as well as detecting the vulnerabilities which should be avoided, it is possible to examine and assess the reliability of the system. Accordingly, methods for improving system reliability are also proposed even though they are mostly only suitable for linear model types.

On the other hand, the concepts of “explainability” and “interpretability” are also successfully separated from each other. By talking about XAI in general, a baseline is created for a systematic overview of different system types revolving this topic, which mainly breaks down to two approaches:

- Transparency models which to some extent could make itself comprehensible
- Black-box machine learning models which require post-hoc techniques to provide an understanding of the models.

While explainability focuses on making the predictions from artificial intelligence systems understandable for human, interpretability has the goal of clarifying the logic behind the models’ algorithms. The main different between explainability and interpretability could be determined based on the type of artificial intelligence models which are being implemented. A transparent model offers a certain degree interpretability since it is easy to analyze the structures and functions of the model itself. Meanwhile, a black-box model requires post-hoc techniques to explain its decisions or predictions, which refers to explainability.

However, a clear and distinct separation between the two terms does not exist, there are always cases where one must combine different methods in order to reach the level of understanding which is desired. In another word, one must stay flexible since there is no pre-defined method that could be applied to every model. This paper simply attempts to sort different models and techniques into specific categories to provide a clearer overview of various approaches which have been researched and developed over the years.

Since XAI is a controversial topic which involves a lot of ethical values as well as responsibilities, different aspects must be guaranteed while implementing an AI system in real-life practice. The most important aspects include fairness, transparency, privacy and safety, some of which could be ensured by applying XAI methods but that does not apply to every aspect.

In conclusion, as AI grows stronger and more important, it is essential to continue researching and developing XAI techniques in order to increase trust in their applications. However, these methods and techniques should also take into consideration the ethical concerns involving the decisions made by AI systems. Only when AI systems are implemented correctly with all things considered could they contribute to the future of technology and mankind.

Bibliography

[1] Adadi, A., & Berrada, M. (2018, September 17). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). From https://ieeexplore.ieee.org/document/8466590

[2] Algoritmi, P., & Cortez, P. (2013, March 01). Using sensitivity analysis and visualization techniques to open black box data mining models. From https://dl.acm.org/doi/10.1016/j.ins.2012.10.039

[3] Arras, L., Montavon, G., Müller, K., & Samek, W. (2017, August 04). Explaining Recurrent Neural Network Predictions in Sentiment Analysis. From https://arxiv.org/abs/1706.07206

[4] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2019, October 22). Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. From https://arxiv.org/abs/1910.10045v1

[5] Auret, L., & Aldrich, C. (2012, June 16). Interpretation of nonlinear relationships between process variables by use of random forests. From https://www.sciencedirect.com/science/article/pii/S0892687512001987

[6] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K., & Samek, W. (2015, July 10). On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. From https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0130140

[7] Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., & Müller, K. (2010, August 01). How to Explain Individual Classification Decisions. From https://dl.acm.org/doi/10.5555/1756006.1859912

[8] Barakat, N., & Bradley, A. (2010, March 27). Rule extraction from support vector machines: A review. From https://www.sciencedirect.com/science/article/pii/S0925231210001591

[9] Barakat, N., & Diederich, J. (2005). Eclectic Rule-Extraction from Support Vector Machines. Retrieved December 1, 2020, from https://core.ac.uk/download/pdf/14982533.pdf

[10] Bastani, O., Kim, C., & Bastani, H. (2018, March 13). Interpretability via Model Extraction. From https://arxiv.org/abs/1706.09773

[11] Beaudouin, V., Bloch, I., Bounie, D., Clémençon, S., D'Alché-Buc, F., Eagan, J., Maxwell, W., Mozharovskyi, P., & Parekh, J. (2020, March 13). Flexible and Context-Specific AI Explainability: A Multidisciplinary Approach. From https://arxiv.org/abs/2003.07703

[12] Boswell, D. (2002, August 6). Introduction to Support Vector Machines. From http://pzs.dstu.dp.ua/DataMining/svm/bibl/IntroToSVM.pdf

[13] Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (2017). *Classification and Regression Trees*. Boca Raton, Florida: Routledge.

[14] Bronshtein, A. (2019, May 06). A Quick Introduction to K-Nearest Neighbors Algorithm. From https://blog.usejournal.com/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7

[15] Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015, August 01). Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission. From https://dl.acm.org/doi/10.1145/2783258.2788613

[16] Che, Z., Purushotham, S., Khemani, R., & Liu, Y. (2015, December 11). Distilling Knowledge from Deep Networks with Applications to Healthcare Domain. From https://arxiv.org/abs/1512.03542

[17] Che, Z., Purushotham, S., Khemani, R., & Liu, Y. (2017, February 10). Interpretable Deep Models for ICU Outcome Prediction. From https://www.ncbi.nlm.nih.gov/pubmed/28269832

[18] Choi, E., Bahadori, M., Kulas, J., Schuetz, A., Stewart, W., & Sun, J. (2017, February 26). RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. From https://arxiv.org/abs/1608.05745

[19] Clark, M. (2019, February 17). Generalized Additive Models. From https://m-clark.github.io/generalized-additive-models/

[20] Cortez, P., & Embrechts, M. (1970, January 01). Opening black box Data Mining models using Sensitivity Analysis: Semantic Scholar. From https://www.semanticscholar.org/paper/Opening-black-box-Data-Mining-models-using-Analysis-Cortez-Embrechts/de03d9438be30501a4d219f50333ac72d3ad4c6d

[21] Datta, A., Sen, S., & Zick, Y. (1970, January 01). Algorithmic Transparency via Quantitative Input Influence. From https://link.springer.com/chapter/10.1007/978-3-319-54024-5_4

[22] Deng, H. (2014, August 23). Interpreting Tree Ensembles with inTrees. From https://arxiv.org/abs/1408.5456

[23] Dilmaghani, S. E., Brust, M., Danoy, G., Cassagnes, N., Pecero, J., & Bouvry, P. (2020, February 24). Privacy and Security of Big Data in AI Systems:A Research and Standards Perspective. From https://orbilu.uni.lu/handle/10993/42478

[24] Domingos, P. (2000, July 24). Knowledge discovery via multiple models. From https://www.sciencedirect.com/science/article/abs/pii/S1088467X98000237

[25] Doran, D., Schulz, S., & Besold, T. (2017, October 02). What Does Explainable AI Really Mean? A New Conceptualization of Perspectives. From https://arxiv.org/abs/1710.00794

[26] Doshi-Velez, F., & Kim, B. (2017, March 02). Towards A Rigorous Science of Interpretable Machine Learning. From https://arxiv.org/abs/1702.08608

[27] Edwards, L., & Veale, M. (2017, December 04). Slave to the Algorithm? Why a 'Right to an Explanation' Is Probably Not the Remedy You Are Looking For. From https://dltr.law.duke.edu/2017/12/04/slave-to-the-algorithm-why-a-right-to-an-explanation-is-probably-not-the-remedy-you-are-looking-for/

[28] “explain.”, “interpret.” Cambridge Dictionary (2020). From: dictionary.cambridge.org

[29] “explain.”, “interpret.” Oxford Learner’s Dictionaries (2020). From: oxfordlearnersdictionaries.com

[30] Fu, X., Ong, C., Keerthi, S., Hung, G. G., & Goh, L. (2005, January 17). Extracting the knowledge embedded in support vector machines. From https://ieeexplore.ieee.org/abstract/document/1379916

[31] Fürnkranz, J., & Kliegr, T. (2015, August 02). A Brief Overview of Rule Learning. From https://link.springer.com/chapter/10.1007/978-3-319-21542-6_4

[32] Gall, R. (2018, December). Machine Learning Explainability vs Interpretability: Two concepts that could help restore trust in AI. From https://www.kdnuggets.com/2018/12/machine-learning-explainability-interpretability-ai.html

[33] Gilpin, L., Bau, D., Yuan, B., Bajwa, A., Specter, M., & Kagal, L. (2019, February 03). Explaining Explanations: An Overview of Interpretability of Machine Learning. From https://arxiv.org/abs/1806.00069

[34] Glasersfeld, E. V. (2002, July 04). On the concept of interpretation. From https://www.sciencedirect.com/science/article/pii/0304422X83900281

[35] Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2014, March 20). Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. From https://arxiv.org/abs/1309.6392

[36] Gopinath, D., Katz, G., Păsăreanu, C., & Barrett, C. (2018, October 07). DeepSafe: A Data-Driven Approach for Assessing Robustness of Neural Networks. From https://link.springer.com/chapter/10.1007/978-3-030-01090-4_1

[37] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Pedreschi, D., & Giannotti, F. (2018, June 21). A Survey Of Methods For Explaining Black Box Models. From https://arxiv.org/abs/1802.01933

[38] Halpern, J. Y., & Pearl, J. (2005, November 7). Causes and Explanations: A Structural-Model Approach. Part I: Causes. Retrieved December 1, 2020, from https://arxiv.org/pdf/cs/0011012

[39] Hara, S., & Hayashi, K. (2017, February 28). Making Tree Ensembles Interpretable: A Bayesian Model Selection Approach. From https://arxiv.org/abs/1606.09066

[40] Hastie, T., & Tibshirani, R. (1986). Generalized Additive Models. From https://projecteuclid.org/euclid.ss/1177013604

[41] Hendrycks, D., & Dietterich, T. (2019, March 28). Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. From https://arxiv.org/abs/1903.12261

[42] Henelius, A., Puolamäki, K., & Ukkonen, A. (2017, July 24). Interpreting Classifiers through Attribute Interactions in Datasets. From https://arxiv.org/abs/1707.07576

[43] Hind, M. (2019, April 01). Explaining explainable AI. From https://dl.acm.org/doi/10.1145/3313096

[44] Holzinger, A., Langs, G., Denk, H., Zatloukal, K., & Müller, H. (2019, February 24). Causability and explainability of artificial intelligence in medicine. From https://pubmed.ncbi.nlm.nih.gov/32089788/

[45] Huang, X., Kwiatkowska, M., Wang, S., & Wu, M. (2017, May 05). Safety Verification of Deep Neural Networks. From https://arxiv.org/abs/1610.06940

[46] Katz, G., Barrett, C., Dill, D., Julian, K., & Kochenderfer, M. (2017, May 19). Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. From https://arxiv.org/abs/1702.01135

[47] Koh, P. W., & Liang, P. (2017, July 10). Understanding Black-box Predictions via Influence Functions. From https://arxiv.org/abs/1703.04730

[48] Krakovna, V., & Doshi-Velez, F. (2016, September 30). Increasing the Interpretability of Recurrent Neural Networks Using Hidden Markov Models. From https://arxiv.org/abs/1606.05320

[49] Lewis, D. (2003, November). Philosophical Papers Volume II. From https://oxford.universitypressscholarship.com/view/10.1093/0195036468.001.0001/acprof-9780195036466

[50] Lipton, Z. (2017, March 06). The Mythos of Model Interpretability. From https://arxiv.org/abs/1606.03490

[51] Lundberg, S., & Lee, S. (2017, November 25). A Unified Approach to Interpreting Model Predictions. From https://arxiv.org/abs/1705.07874

[52] Maaten, L. V., & Hinton, G. (2008, November). Visualizing Data using t-SNE. Retrieved December 1, 2020, from http://www.cs.toronto.edu/~hinton/absps/tsnefinal.pdf

[53] Magdalena, L. (1970, January 01). Fuzzy Rule-Based Systems. From https://link.springer.com/chapter/10.1007/978-3-662-43505-2_13

[54] Mahendran, A., & Vedaldi, A. (2014, November 26). Understanding Deep Image Representations by Inverting Them. From https://arxiv.org/abs/1412.0035

[55] Medsker, L. R., & Jain, L. C. (2000). *Recurrent neural networks design and applications*. Boca Raton, Florida: CRC Press.

[56] Minsky, M., & Papert, S. (1969, January). Perceptrons: An Introduction to Computational Geometry. From https://mitpress.mit.edu/books/perceptrons

[57] Molnar, C. (2020, November 30). Interpretable Machine Learning. From https://christophm.github.io/interpretable-ml-book/

[58] Montavon, G., Bach, S., Binder, A., Samek, W., & Müller, K. (2015, December 08). Explaining NonLinear Classification Decisions with Deep Taylor Decomposition. From https://arxiv.org/abs/1512.02479

[59] Navia-Vázquez, A., & Parrado-Hernández, E. (2006, February 20). Support vector machine interpretation. From https://www.sciencedirect.com/science/article/abs/pii/S0925231205004480

[60] Navlani, A. (2018, August 2). KNN Classification using Scikit-learn. From https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

[61] Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., & Clune, J. (2016, December 01). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. From https://dl.acm.org/doi/10.5555/3157382.3157477

[62] Núñez, H., Angulo, C., & Català, A. (2002). Rule extraction from support vector machines. From https://www.academia.edu/849461/Rule_extraction_from_support_vector_machines

[63] Núñez, H., Angulo, C., & Català, A. (2002, February). Support vector machines with symbolic interpretation. From https://ieeexplore.ieee.org/document/1181456

[64] Núñez, H., Angulo, C., & Català, A. (2006). Rule-Based Learning Systems for Support Vector Machines. From https://link.springer.com/content/pdf/10.1007/s11063-006-9007-8.pdf

[65] Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., & Mordvintsev, A. (2020, January 10). The Building Blocks of Interpretability. From https://distill.pub/2018/building-blocks/

[66] Oliveira, T., Barbar, J., & Soares, A. (2014, September 18). Multilayer Perceptron and Stacked Autoencoder for Internet Traffic Prediction. From https://link.springer.com/chapter/10.1007/978-3-662-44917-2_6

[67] O'Shea, K., & Nash, R. (2015, December 02). An Introduction to Convolutional Neural Networks. From https://arxiv.org/abs/1511.08458v2

[68] Papernot, N., & McDaniel, P. (2018, March 13). Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning. From https://arxiv.org/abs/1803.04765

[69] Park, S., & Han, K. (2018, March). Methodologic Guide for Evaluating Clinical Performance and Effect of Artificial Intelligence Technology for Medical Diagnosis and Prediction. From https://pubmed.ncbi.nlm.nih.gov/29309734/

[70] Ribeiro, M., Singh, S., & Guestrin, C. (2016, August 09). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. From https://arxiv.org/abs/1602.04938

[71] Ribera, M., & Lapedriza, A. (2019). Can we do better explanations? A proposal of User-Centered Explainable AI. Retrieved December 1, 2020, from http://ceur-ws.org/Vol-2327/IUI19WS-ExSS2019-12.pdf

[72] Robnik-Šikonja, M., & Kononenko, I. (2008, May 01). Explaining Classifications For Individual Instances. From https://dl.acm.org/doi/10.1109/TKDE.2007.190734

[73] Ronan, H., Henrik, J., & Ignacio, S. M. (2020, January 14). Robustness and Explainability of Artificial Intelligence. From https://ec.europa.eu/jrc/en/publication/robustness-and-explainability-artificial-intelligence

[74] Roscher, R., Bohn, B., Duarte, M. F., & Garcke, J. (2020, January 12). Explainable Machine Learning for Scientific Insights and Discoveries. From https://arxiv.org/abs/1905.08883

[75] Rosenbaum, L., Hinselmann, G., Jahn, A., & Zell, A. (2011, March 25). Interpreting linear support vector machine models with heat map molecule coloring. From https://pubmed.ncbi.nlm.nih.gov/21439031/

[76] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. From http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.335.3398

[77] Samek, W., Wiegand, T., & Müller, K. (2017, August 28). Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. From https://arxiv.org/abs/1708.08296

[78] Sato, M., & Tsukimoto, H. (2002, August 7). Rule extraction from neural networks via decision tree induction. From https://ieeexplore.ieee.org/abstract/document/938448

[79] Scott, A., Clancey, W., Davis, R., & Shortliffe, E. (1977, February). Explanation Capabilities of Production-Based Consultation Systems. From https://www.aclweb.org/anthology/J77-1006/

[80] Shrikumar, A., Greenside, P., & Kundaje, A. (2019, October 12). Learning Important Features Through Propagating Activation Differences. From https://arxiv.org/abs/1704.02685

[81] Strumbelj, E., & Kononenko, I. (2010, March 01). An Efficient Explanation of Individual Classifications using Game Theory. From https://dl.acm.org/doi/10.5555/1756006.1756007

[82] Su, G., Wei, D., Varshney, K., & Malioutov, D. (2016, June 18). Interpretable Two-level Boolean Rule Learning for Classification. From https://arxiv.org/abs/1606.05798

[83] Tan, S., Caruana, R., Hooker, G., & Lou, Y. (2018, October 11). Distill-and-Compare: Auditing Black-Box Models Using Transparent Model Distillation. From https://arxiv.org/abs/1710.06169

[84] Tolomei, G., Silvestri, F., Haines, A., & Lalmas, M. (2017, June 20). Interpretable Predictions of Tree-based Ensembles via Actionable Feature Tweaking. From https://arxiv.org/abs/1706.06691

[85] Valenzuela-Escárcega, M., Nagesh, A., & Surdeanu, M. (2018, May 29). Lightly-supervised Representation Learning with Global Interpretability. From https://arxiv.org/abs/1805.11545v1

[86] Vehicle Automation Report HWY18MH010. From: National Transportation Safety Board Office of Highway Safety Washington, D.C. (2019)

[87] Waltl, D., Bonczek, G., & Matthes, F. (2018). Rule-based Information Extraction - Advantages, Limitations, and Perspectives. From https://wwwmatthes.in.tum.de/pages/1w12fy78ghug5/Rule-based-Information-Extraction-Advantages-Limitations-and-Perspectives

[88] Wang, T. (2019, May 10). Gaining Free or Low-Cost Transparency with Interpretable Partial Substitute. From https://arxiv.org/abs/1802.04346

[89] Wenzel, F., Galy-Fajou, T., Deutsch, M., & Kloft, M. (2017, July 18). Bayesian Nonlinear Support Vector Machines for Big Data. From https://arxiv.org/abs/1707.05532

[90] Wiese, H. (2010, April). Applied Cooperative Game Theory: The Gloves Game. From https://www.wifa.uni-leipzig.de/fileadmin/user_upload/itvwl-vwl/MIKRO/Lehre/CGT-applications/gl.pdf

[91] Wisdom, S., Powers, T., Pitton, J., & Atlas, L. (2016, November 22). Interpretable Recurrent Neural Networks Using Sequential Sparse Recovery. From https://arxiv.org/abs/1611.07252v1

[92] Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y., & Zhang, Z. (2014, November 24). The Application of Two-level Attention Models in Deep Convolutional Neural Network for Fine-grained Image Classification. From https://arxiv.org/abs/1411.6447

[93] Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., & Zhu, J. (2019, October 09). Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges. From https://link.springer.com/chapter/10.1007/978-3-030-32236-6_51

[94] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., & Bengio, Y. (2016, April 19). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. From https://arxiv.org/abs/1502.03044

[95] Yang, C., Rangarajan, A., & Ranka, S. (2018, May 23). Global Model Interpretation via Recursive Partitioning. From https://arxiv.org/abs/1802.04253

[96] Zeiler, M., Krishnan, D., & Taylor, G. W. (2010). Deconvolutional Networks. Retrieved December 1, 2020, from https://www.matthewzeiler.com/mattzeiler/deconvolutionalnetworks.pdf

[97] Zeiler, M., Taylor, G., & Fergus, R. (2011, November 01). Adaptive deconvolutional networks for mid and high level feature learning. From https://dl.acm.org/doi/10.1109/ICCV.2011.6126474

[98] Zhang, Q., Wu, Y., & Zhu, S. (2018, February 14). Interpretable Convolutional Neural Networks. From https://arxiv.org/abs/1710.00935

[99] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning Deep Features for Discriminative Localization. From https://ieeexplore.ieee.org/document/7780688

[100] Zilke, J. R., Mencía, E. L., & Janssen, F. (2016). DeepRED – Rule Extraction from Deep Neural Networks. Retrieved December 1, 2020, from https://www.ke.tu-darmstadt.de/publications/papers/DS16DeepRED.pdf

[101] Üstün, B., Melssen, W., & Buydens, L. (2007, March 18). Visualisation and interpretation of Support Vector Regression models. From https://www.sciencedirect.com/science/article/abs/pii/S0003267007004904

Statement of Authorship

I hereby declare according to the examination regulations §16 (5) APSO-TI-BM/APSO-INGI, that I am the sole author of this bachelor thesis and that I have not used any sources other than those listed in the bibliography and identified as references. I further declare that I have not submitted this thesis at any other institution in order to obtain a degree.

Hamburg, den _____ _____