Universiteit Leiden
The Netherlands

Opleiding Informatica & Economie

Explainable Artificial Intelligence:

a Checklist for the Insurance Market

Olivier Koster

University supervisor:
Joost Visser

Company supervisors:
Ruud Kosman & Walter Mosterd

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

07/03/2021

Abstract

Artificial intelligence (AI) can be a powerful tool to accomplish a great many tasks. This exciting branch of technology is being adopted increasingly across varying sectors, including healthcare and insurance. With that potential often arise several complications. One of which is a lack of transparency and explainability of an algorithm for experts and non-experts alike. This brings into question both the usefulness as well as the accuracy of the algorithm, coupled with an added difficulty to assess potential biases within the model. In this thesis, we investigate the current usage of AI algorithms in the Dutch insurance industry and the adoption of explainable artificial intelligence (XAI) techniques. Armed with this knowledge we design a checklist for insurance companies that should help assure quality standards and a solid foundation for co-operation between firms. This checklist extends an existing checklist that SIVI, a knowledge and standardisation institute for financial services, offers its member organisations.

Contents

1 Introduction

1.1 Background and motivation

Artificial Intelligence (AI) is one of the leading technologies paving the way for more efficient solutions and powerful automation. This exciting technology is being deployed increasingly across various industries. For instance, AI is aiding healthcare in its search for accurate diagnostic procedures in order to detect cancer early, assisting radiology by discovering patterns and accelerating medicine development [Daley, 2019]. Similarly, in the insurance industry, the use of AI is starting to gain traction, being used in assessing risks, handling claims and detecting fraud.

Aside from all this added ability, AI too comes with its own downsides. Much like human cognition, technology has its flaws. The same goes for AI-algorithms. While early AI systems were relatively easy to comprehend, we have seen a recent rise in opaque decision systems such as Deep Neural Networks (DNNs). Although these types of algorithms increase accuracy, they come with a higher level of algorithmic complexity, often consisting of hundreds of layers and millions of parameters [Castelvecchi, 2016]. In these instances, interpretability vastly decreases. This results in a black-box algorithm which is difficult to understand for experts and non-experts alike.

A black-box is an abstract device where, given a set of known inputs and outputs, we do not know the inner workings of the device. This can create some unfavourable situations. In particular, when decisions made by an algorithm affect human lives. For example, when black-box algorithms make incorrect diagnoses, doctors may be subject to intense scrutiny for taking the wrong course of action, being unable to explain proper reasoning behind a diagnosis.

This phenomenon can have an even more severe impact on a larger scale. In 2020, The Dutch government deployed SyRi, an algorithmic fraud risk scoring system. It used a non-disclosed algorithmic risk model to profile citizens, allegedly targeting mostly low-income neighbourhoods and minority residents. Dutch court deemed SyRi illegal for lacking transparent data usage and violating privacy. Similarly, Apple revealed its AI-driven credit risk system showed a striking bias, as it deemed men far more creditworthy than women. Consequently, individuals could be given different credit limits despite having the same accounts, cards or assets [Kroes, 2020].

In precision medicine, decisions cannot be based on mere binary prediction, creating a need for extensive explanations supporting a models output. The same holds true for other industries such as autonomous vehicles, transportation, security, and finance among others [Tjoa and Guan, 2020].

Evidently, using algorithms to make impactful decisions can be a rather dangerous practice when the legitimacy of the model is not justified. From this perspective, the responsibility of the algorithms' creator does not only concern its accuracy but also with its interpretability and transparency. This reasoning has spawned a new field of research named Explainable Artificial Intelligence (XAI).

Research on this topic has spiked in recent years, reflecting a growing need for XAI. Even so, to our knowledge, no prior research has been done that captures the state of AI adoption, and more specifically of XAI techniques, in the (Dutch) insurance industry.

1.2 Thesis outline

In this thesis, we examine the current adoption, as well as future prospects, of AI and XAI within the Dutch insurance industry. This is done through rigorous literature research coupled with conducting exploratory interviews of industry experts. All of this will lay the groundwork for the design of a checklist for insurance companies that should help assure quality standards and a solid foundation for co-operation between firms. The checklist is then evaluated and tested by conducting confirmatory interviews, creating a feedback loop for further refinement. The aforementioned checklist extends an existing checklist that SIVI, a knowledge and standardisation institute for financial services, offers its member organisations. The research methodology is based on the design science research paradigm for information systems [Hevner et al., 2004].

1.3 Research questions

In order to effectively design our checklist, we consult literature and industry expertise to answer the following research questions:

"How can we assess the efforts taken to increase model transparency and increase its explainability in the organisational environment, making sure all practical and ethical facets are taken into consideration?"

This question can be tackled by answering the following subquestions:

1. "What are the factors that impact model transparency and explainability?"
2. "How do model transparency and explainability impact its related processes and stakeholders?"

All research is conducted under the supervision of Joost Visser, professor of large Scale Software and Data Science at Leiden University, and Walter Mosterd and Ruud Kosman both Consultants at SIVI.

2 Theoretical framework

In this chapter, we lay out the relevant literature within the field to act as a knowledge base for the design of our checklist. We dive into the most important, basic concepts and terminology of XAI, as well as break down the factors that explainability is influenced by. From there we split the field up into specific subdomains, namely transparent and non-transparent models, explore their respective characteristics and identify possible stakeholders. We conclude with reviewing the implications explainability can have and breaking down the different methods to overcome these obstacles.

2.1 Taxonomy of Explainable Artificial Intelligence (XAI)

In this section, we cover all relevant factors that attribute to or are influenced by transparency and explainability.

2.1.1 Terminology

In this subsection, we cover all relevant terminology used in the current field of Explainable Artificial Intelligence (XAI). According to Vilone and Longo [2020] there is little agreement among scholars on what explanations are and what properties they might have, as well as the correct terminology that should be used.

Fortunately, Barredo Arrieta et al. [2020] have comprised a clear report on the most common terms used in XAI research and their respective meaning. Below the literal definitions are listed from that article.

- *Understandability* (or equivalently, *intelligibility*) denotes the characteristic of a model to make a human understand its function – how the model works – without any need for explaining its internal structure or the algorithmic means by which the model processes data internally
- *Comprehensibility*, when conceived for ML models, refers to the ability of a learning algorithm to represent its learned knowledge in a human-understandable fashion. Given its difficult quantification, comprehensibility is normally tied to the evaluation of the model complexity.
- *Interpretability* is defined as the ability to explain or to provide the meaning in understandable terms to a human.
- *Explainability* is associated with the notion of explanation as an interface between humans and a decision-maker that is, at the same time, both an accurate proxy of the decision-maker and comprehensible to humans.
- *Transparency*: A model is considered to be transparent if by itself it is understandable. Since a model can feature different degrees of understandability, transparency, a term we will discuss more in-depth next, is divided into three categories: Simulatability, decomposability and algorithmic transparency. Each of these classes contains its successor (e.g. a simulatable model is at the same time a model that is decomposable and algorithmically transparent).

The respective meaning of these terms used in the definition of transparency are listed below:

- *Simulatability*: The degree to which a model is able to be simulated or thought about strictly by a human.

- *Decomposability*: The degree to which a model can be decomposed into its individual components (input, parameters and output), where each component is individually interpretable;
- *Algorithmic transparency*: The degree of confidence of a learning algorithm to behave 'sensibly' in general

For simplicity, we will use only two main definitions in this paper to describe the black-box problem. We will use the term **transparency** (and its subcategories) to denote a model's innate understandability, how well it conveys its inner workings to the decision-maker. This allows us to express a model as transparent or oppositely, opaque. We will go into greater detail about the relevance of this later.

We use the term transparency because it goes hand in hand with the intuitive description of a black-box. When imagining a black-box we see the black outer layer as that which is blocking us from seeing the inner components of the box, meaning the outer shell is completely opaque. When increasing understandability the outer shell becomes more transparent to the point where we can truly see all of its components and deduce they're individual relations.

We will use the term **explainability** to denote the way a model conveys, in a human-understandable way, how the input leads to the results (i.e. output), which in turn can lead to novel or confirming insights (about the model and its dataset). As opposed to only giving insight into the model itself, increased explainability aims to deepen the understanding of the modelled topic. Thus, potentially, answering the 'why' question, instead of the 'what' question AI-models usually seek out to answer. An illustrative example of explainability is given in Section 2.2.

To get a better grip on the concept of explaining AI-models, we must acknowledge that there are many possible types of explanation and define the formation of several attributes and structures. To organise these, similar to Vilone and Longo [2020], we propose the following clusters; Attributes of explainability and types of explanation.

2.1.2 Attributes of explainability

There are several attributes to explainability. In this subsection, we will talk about the purposes of explainability, explanation requirements, and characteristics of explainability.

There are multiple reasons why we might need to explain a model. The foremost and arguable most obvious reason would be to *improve* the model. A deep understanding of a model can act as insurance that only meaningful variables infer the output. This is especially helpful when, for example, determining feature relevance beforehand. Often, AI algorithms determine correlation very differently from a human-centred way of thinking. Humans typically think in terms of cause-and-effect, whereas computers have no such limitations to start with. In that regard, AI can support the extraction of novel information to us humans by way of explanations. As such, explanations can help us *discover* new correlations that are not previously evident to us. Of course, we would want these correlations to be accurate and supported with sensible explanations. especially in a world where there is a growing demand for ethical AI. This spawns another reason for explainability, which is to *justify* our model. In other words, we would want to prove the robustness of the model as well as highlight potential adversarial perturbations that could change the prediction or any potential impartiality and bias that could result in any ethical concerns. Lastly, explanations help us *control* the correct functioning of our model and allow us to debug and highlight potential flaws. [Vilone and Longo, 2020; Barredo Arrieta et al., 2020]

To create meaningful explanations that serve the aforementioned purposes, Vilone and Longo [2020] propose a set of base requirements that, when followed properly, should make for the most complete and informative explanations. These requirements are listed below without modification:

- *Fidelity*: the representation of inputs and models in terms of concepts should preserve and present to end-users their relevant features and structures.
- *Diversity*: inputs and models should be representable with few non-overlapping concepts.
- *Grounding*: concepts should have an immediate human-understandable interpretation.
- *Graphical integrity*: the representations should highlight the features that contribute the most to the final predictions and distinguish those with positive and negative attribution
- *Coverage*: a large fraction of the most important features should be visible in the representation
- *Morphological clarity*: the important features should be clearly displayed, their visualisation cannot be 'noisy'
- *Layer separation*: the representation cannot occlude the raw image which should be visible for human inspection.
- *Input invariance*: meaning that a method for explainability must mirror the sensitivity of the underlying model with respect to transformations of the inputs in order to ensure a reliable interpretation of their contribution to each prediction.
- *Implementation invariance*: This suggests that a method applied to functionally equivalent neural networks should assign identical contributions to the features of the input.

Admittedly, according to Vilone and Longo [2020], there are a few characteristics to explanations that should be taken into account. Firstly, explanations have an *Contrastive nature*. That is, people seek for an explanation when they are presented with counterfactual and/or counter-intuitive events. Contrarily, when a model adheres to the beliefs of the explainer and explainee, they might not seek out further explanation. This is due to the *social nature* of explanations as they are part of a dialogue aiming at transferring knowledge. Furthermore, due to a *selectivity of explanations*, people usually do not expect that an explanation contains the actual and complete list of the causes of an event, but only a selection of the few causes deemed to be necessary and sufficient to explain it. Authors point out the risk that this selection might be influenced by cognitive biases. Lastly, we see an *irrelevance of probabilities to explanations*, referring to the occurrence probabilities of events or the statistical relationships between causes and events does not produce a satisfactory and intuitive explanation. Explanations are more effective when they refer to the causes and not to their likelihood.

2.1.3 Types of explanations

As with most things in life, there isn't one single best way to do anything. The same holds true for explaining AI-models. There are many ways to explain a complex model. In our case, we lay out the many possible types of explanations and their characteristics. By doing this we create the possibility to classify an explanation method as a certain type. This is helpful because it will later allow us to order the different explaining methods that have been developed until now according to there practical usability.

Firstly, in accordance with the categories proposed by Haynes et al. [2009] and Sheh and Monteath [2017], we can categorise explanations according to the specific goal they aim to satisfy. This reasoning establishes the following types:

- *Traced-based* or *mechanistic explanations*: Offer insight into the way model components interact and their cause-and-effect relations. These types of explanations are useful for system designers, that accurately reflects the reasoning implemented within a model. They are central to understanding the reasoning of a model.
- *Reconstructive* or *ontological explanations*: is designed for end-users and has a problem-solving oriented approach. In a sense, a typical reconstructive explanation builds a story for a specific case, exposing the input features contributing to the prediction. For instance, an image of a bird was assigned to a certain class because of the colour of the bird. However, the model might have analysed other features that did not influence the final assessment, like the image's background. These characteristics can be included in the traced-based explanations but excluded from the reconstructive explanations.
- *Operational explanations*: Tells a user which components will contribute to the goals that the model is designed to achieve. Telling the user "How do I use the model?".
- *Teaching explanations*: Aims at informing humans about the concepts learned by the system such as, for example, the presence of some physical constraints (walls or other obstacles) that can limit its actions.
- *Introspective tracing explanations*: Have the goal of finding the cause of/and the solution to a fault.
- *introspective informative explanations*: Aim at explaining predictions based on the reasoning process to improve human-system interaction
- *Postmodum explanations*: Focused on explaining the decisions directly linking them with the inputs.
- *Execution explanations*: Explanations that directly follow the exact execution of the system.
- *Reasoning domain knowledge*: Focused on the domain knowledge needed to perform reasoning, including rules and terminology.
- *Communication domain knowledge*: Focused on how to communicate domain knowledge and deals with practical aspects of the communication process (i.e. the language to be used, the most effective strategies for effective explanations and the communication medium). This knowledge must be tuned to the knowledge of the specific audience in mind.

Various other characteristics can be used to further categorise the various explanation methods. When referring to the scope of a method, we say the explanation is either *global*, meaning the explanation takes all instances of input data to make the entire model comprehensible as a whole, whereas *local* explanations only make a single explicit inference of a model, meaning speaking about a specific instance of data. Then, explanations can occur at different stages in the model's process. *Ante-hoc* (or sometimes referred to as *ex-ante*) explanations are techniques that aim to increase explainability from the beginning to the end of all preparation stages (i.e. input normalisation, feature selection, or in the case of ML, training the model). *Post-hoc* explanations are those techniques that occur at run time or use the model's output itself. We can further divide the category of Post-hoc methods into*model-agnostic* and *model-specific* methods. The

former referring to methods that apply to all kinds of algorithms and the latter referring to methods that are only applicable for specific types of algorithms (e.g. specifically for Support Vector Machines).

There is another constraint that determines the type of method. For example, time-series data is used by different methods than image data. Thus, the input that the model receives dictates the type of explanation needed. Later we will propose methods that take either, *numerical/categorical*, *pictorial*, *textual* or *time series* as inputs. Contrarily, post-hoc methods can result in different kinds of output formats, although not all methods are limited to a single type of output format. So far we have only seen methods that result in *numerical explanations*, *rule-based explanations*, *textual explanations*, *visual explanations*, *mixed explanations*. Often, the output format type most suitable for the specific circumstance is chosen.

2.1.4 Transparent models

By now we have established that there is a disparity between accuracy and transparency in AI-models. Where gains are made in one, returns diminish for the other. Figure 1, by Barredo Arrieta et al. [2020], illustrates this trade-off graphically.

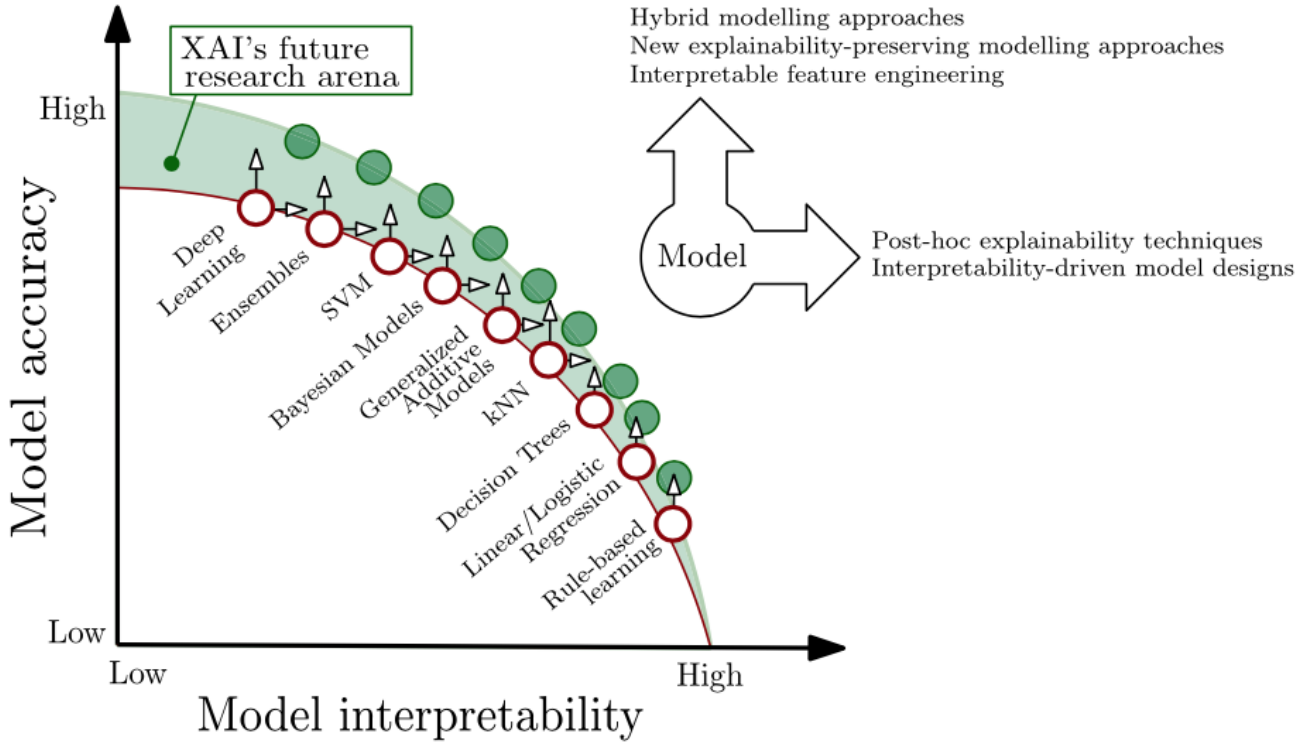

Figure 1: Model Accuracy/transparency trade-off
[Barredo Arrieta et al., 2020]

As we argued in Section 2.1.1 transparent models are models where one can view its individual components and deduce how they relate to each other to create the output from a given input. In that sense, we call a model a transparent model when it is understandable by itself and no other external explanations are

required, although ex-ante and post-hoc techniques can often still be used to increase explainability regardless.

In the field of AI, we divide the different sub-fields into two categories that relate to their respective level of transparency: *transparent models* and *non-transparent* models. Generally, transparent models fare well in most of the three levels of transparency discussed earlier. Thus, we can determine if a model is transparent based on these three characteristics. Non-transparent models do not satisfy any of the three forms well. If you recall, *simulatability* is the degree to which a model is able to be simulated or thought about strictly by a human, *decomposability* is the degree to which a model can be decomposed into its individual components (input, parameters and output), where each component is individually interpretable, and *algorithmic transparency* is the degree of confidence of a learning algorithm to behave 'sensibly' in general. For every transparent model, we will explain how they fare with respect to every form of transparency, according to Barredo Arrieta et al. [2020]:

- *Rule-Based Systems (RBS)*: A RBS is a system that applies human-made rules to store, sort and manipulate data. To work, RBSs require a set of facts or source of data, and a set of rules for manipulating that data. These rules are sometimes referred to as 'If statements' as they tend to follow the line of 'IF X happens THEN do Y'. Rules are one of the more intuitive ways to convey information to humans. That is, unless they use too many rules, thus, becoming too complex for a human to effectively comprehend. This is why RBSs are only simulatable, when the amount of rules is not too extensive, creating the need for additional post-hoc analysis. They are, however, generally decomposable, in that every rule can be separately inspected and individual rules give information about the some of their parts. They are also algorithmically transparent, as rules are human-readable, explain the knowledge learned from data and allow for a direct understanding of the prediction process. This can notably also be the case for non-experts.

- *linear/logistic regression*: Logistic regression is a supervised learning classification model which predicts a discrete binary dependent variable (i.e. target value). However, when the dependent variable is continuous, linear regression would its proper name. This model operates under the assumption that the predictors and the predicted variables are linearly dependent, impeding a flexible fit to the data. This inherent 'stiffness' is the main reason these models can be considered transparent models. Predictors are human-readable and interactions among them are decomposable. This also makes for good simulatability. However, explainability is can also be geared towards non-experts, which makes the model fall under both categories depending on who is to interpret it. They are generally only algorithmically transparent when paired with mathematical tools. Although they meet the characteristics of transparent models, they may also demand post-hoc explainability techniques for visualisation, when the model is to be explained to non-expert audiences.

- *Decision tree learners*: Decision tree (supervised) learning uses a decision tree as a predictive model to go from observations about an item, represented in the branches, to conclusions about the item's target value, represented in the leaves. This simplicity gives decision trees off-the-shelf transparency, making them among the most popular machine learning algorithms. Tree models where the target variable is discrete are called classification trees. Decision trees where the target variable can take continuous values are called regression trees. In their simplest forms (i.e. its size is somewhat small and the amount of features and their meaning are easily understandable), decision trees are simulatable. A human can simulate and obtain the prediction of a decision tree, without requiring any mathematical background. However, their properties can render them decomposable

or algorithmically transparent. An increment in size transforms the model into a decomposable one since its size impedes its full simulation by a human. Further increasing its size and using complex feature relations will make the model algorithmically transparent, losing the previous characteristics.

- *K-nearest neighbours*: K-nearest neighbours (k-NN) is another supervised learning model. This type of model predicts the class of a test sample by a plurality vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours, where the neighbourhood relation is induced by a measure of distance between samples. When used in the context of regression problems, it predicts the property value for the object and the voting is replaced by an aggregation (e.g. average) of the target values associated with the nearest neighbours. Interestingly, this prediction approach resembles that of experience-based human decision making, which decides upon the result of past similar cases. This characteristic explains why k-NN has also been adopted widely when transparency is deemed an important factor. A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data. The k-NN model's class of transparency depends on the features, the number of neighbours and the distance function used to measure the similarity between data instances. A very high k hinders complete simulation of the model by a human user, but k-NN otherwise simulatable. Similarly, the distance function and the set of variables can be decomposable and separately analysed, but the usage of complex features or distance functions would hamper the model's decomposability. Furthermore, increasing the number of variables so much that the user has to rely on mathematical and statistical tools to analyse the model, would make the model merely algorithmically transparent.

- *Rule-based learners*: Rule-based learners are supervised learning prediction models that, similarly to RBSs, use rules to manipulate data. The key difference being that these rules are generated by the model and not a human. They are also structurally related to decision trees as rules can be converted to trees and vice versa. Because rule-based learners have an operational similarity to human rationale, they are often used to explain other complex models by generating rules that explain their predictions. This makes them especially suitable for knowledge representation in expert systems. If the amount of rules is manageable by a human user without external methods, such a model is simulatable. However, problems with these models occur with significant increases in the number and length of the rules generated. If this is the case the model becomes only decomposable. The greater the number or length of rules, the closer the model will be to being just algorithmically transparent. Even so, if a certain threshold of coverage is acquired, a rule-wrapper can sometimes be thought to contain enough information about a model to explain its behaviour to a non-expert user.

  Rules can also be represented with a non-binary truth value, which can be any real number between 0 and 1. We call this subcategory of models *fuzzy-based learners*. They are designed for a broader scope, allowing for the definition of verbally formulated rules over imprecise domains. They perform better than classical rules in situations with higher degrees of uncertainty, whilst enhancing models since they operate in linguistic terms. A particular reason to use fuzzy rules, instead of classical rules, is to diminish the constraints of rule sizes, since a greater range can be covered with less stress on explainability.

- *Generalized additive models*: A generalized additive model (GAM) is a generalized linear model in which a response variable depends linearly on unknown smooth functions of some predictor variables. Such a model infers the smooth functions whose aggregate composition approximates the predicted variable. This empowers the user to examine the effect of variables on the predicted output,

making it readily explainable (with visualisation tools). Thus, GAMs are mainly used to understand relationships between the variables in the dataset, rather than predict outcomes with certain accuracy. Which is why this type of model is often applied in the field of risk assessment. But also finance, environmental studies, geology, healthcare, biology and energy. In its defined form, GAMs can be considered simulatable and decomposable, although additional modification can push this to some extent. Such modifications include the introduction of link functions to relate the aggregation with the predicted output, or the consideration of interactions between predictors.

- *Bayesian models*: Bayesian statistical models use Bayes' theorem to compute and update probabilities after obtaining new data. The theorem describes the conditional probability of an event based on data as well as prior information or beliefs about the event or conditions related to the event. Bayesian models usually express conditional dependencies between variables with a probabilistic directed acyclical graph. For example, a Bayesian network can be used to compute probabilities of diseases given a set of symptoms. Thus, express relationships between symptom and disease. In this way, they share GAMs capabilities to understand relationships between features and their target, now in the form of vertices in a graph rather than smooth functions. Bayesian models are generally understandable to their target audience (experts) and are therefore simulatable as long as variables are kept to a minimum. When the statistical relationships involve too many variables they must be decomposed into marginals to be directly understandable by these experts. If, when decomposed, predictors are still too complex to be understood without mathematical tools, the model loses its status as decomposable as well, becoming just algorithmically transparent.

2.1.5 Non-transparent models

Contrarily, *non-transparent models* (or opaque models) fail to fall under either level of transparency, because of their inherent complexity. This creates a need for ante-hoc and post-hoc techniques. For every model type, we will go over their external method needs, according to Barredo Arrieta et al. [2020].

- *Tree ensembles*: An ensemble method is a supervised learning technique that combines several base models to produce one optimal predictive model. Tree ensembles, otherwise known as random forests, combine several decision trees to produce better predictive performance, than utilising just a single decision tree. The main principle behind the ensemble model is that a group of weak learners come together to form a strong learner. While tree ensembles are overall more complex than single decision trees, they are among the most accurate AI models in use nowadays. Unfortunately, however, the combination of decision trees loses every transparent property, creating a need for ante-hoc and post-hoc techniques. Specifically they usually require model simplification or feature relevance techniques, like *DT extraction*, *FAB*, *TSP*, *Discriminative Patterns*, as mentioned in Vilone and Longo [2020]. In some cases, GAMs can be also used to help explain tree ensembles by averaging them.

- *Support Vector Machines*: A support vector machine (SVM) is a learning method with associated learning algorithms that analyse data for classification and regression analysis or other tasks (e.g. outlier detection). It does this by formulating a (set of) hyper-plane(s) in a high (or infinite) dimensional space. It can have both supervised and unsupervised applications. They are one of the most used ML models, because of their accurate prediction and generalisation capabilities. However, due to their opaque, mathematical structure, SVMs are even more complex than tree ensembles. For

this reason, post-hoc explainability is often aimed at explaining its mathematical internal structure. These techniques (e.g. ExtractRule, SVM+Prototypes, Weighted Linear Classifier), as found in Vilone and Longo [2020], cover model simplification, local explanation techniques, and visualisation techniques in general.

- *Neural networks*: An (artificial) neural network (NN) is a learning method that loosely resembles the biological neural networks found in animal brains. They are a collection of connected nodes, called (artificial) neurons. A NN consist of at least three layers of neurons (an input layer, one or more hidden layers and an output layer). NNs, otherwise known as *Deep Learning*, come in many different shapes, but we will explicitly discuss three types here. Namely, multi-layer neural networks, convolutional neural networks and recurrent neural networks.

  The base type, multi-layer neural networks, formally known as multi-layer perceptrons (MLP), employs a supervised learning technique, called backpropagation, to train on the input data, with non-linear activation functions for each node. Neurons and edges typically have weights that adjust as learning proceeds. The weight increases or decreases the strength of the signal at a connection. By doing this, NNs can distinguish data that is not linearly separable. This ability to infer complex relations among variables gives NNs a broad scope of application. Unfortunately, due to their black-box nature, many are reluctant to adopt NNs, as explainability is often of high practical value. Whereas a single perceptron neural network falls within the class of simulatability, a multi-layer neural network does not fall within either level of transparency. This problem, however, has spawned a lot of research into explainable techniques for NNs.

  A convolutional neural network (CNN or ConvNet), also known as shift invariant or space invariant artificial neural network (SIANN), are regularised versions of MLPs, that use tensors as input (as opposed to vector inputs used by MLPs). Where an MLPs use weights for regularisation, CNNs reorganise hierarchical data patterns into smaller and simpler patterns that ultimately yield a higher collective complexity (a mathematical process known as *convolution*). This way, a lot of parameters that are usually hand-engineered in MLPs, are now learned. This complex structure, however, is very difficult to explain. Even so, their needs for explainability usually revolve around the visualisation, as CNNs are mostly applied to image and video related problems (although they are occasionally used for other purposes e.g. natural language processing or financial time series).

  A recurrent neural network (RNN), like its name suggests, are a class of neural networks that allow previous outputs to be used as inputs. This makes it especially suitable for sequential data or time-series data. These types of data exhibit long-term dependencies that are complex to be captured by a ML model. RNNs are able to store information about these time-dependent relationships, by use of so-called 'hidden states'. Consequently, RNNs are commonly used for ordinal or temporal problems (e.g. language translation, natural language processing, speech recognition, and image captioning). This, however, creates an extra layer of complexity, with regard to human comprehension.

  NN explainability mostly revolves around model simplification, feature relevance, local explanation or visualisation techniques. As mentioned, such NN explainability techniques can be found in Vilone and Longo [2020].

2.1.6 Stakeholders

This section takes inspiration from the stakeholders and goals explained by Barredo Arrieta et al. [2020]; Tomsett et al. [2018]; Preece et al. [2018]; Zednik [2019] to combine their perspectives.

To understand the concept of explainability, we must recognise that opacity is agent-relative. That is to say, a system itself is never opaque but rather opaque with respect to some particular agent

Depending on what a particular agent is tasked with doing, they are likely to require a different kind of knowledge to do it and, thus, seek a different kind of explanation. But who are these stakeholders, and what exactly are they tasked with doing?

Several articles on this topic have differing opinions of the possible stakeholder groups that can be defined. Interestingly, these differences are mostly caused by the contrasting perspectives from which we can view the XAI-domain. In our opinion, the best starting point would be to assess all possible agents that may interact with or would be subject to the system. From there, we can examine the different demands they might have concerning explanations. Tomsett et al. [2018] break down all different agent groups within a system in fine detail. We slightly modified the figure they use, which can be seen in figure 2, below.

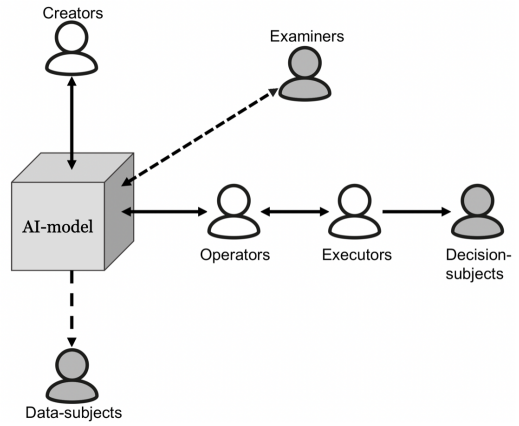

Figure 2: Six model agents
derived from an image from Tomsett et al. [2018]

- *Creators*: Agents that create the system. Several different types of personnel may fall into this group. Importantly, we can subdivide this group further into what we might call the *owners* and the *implementers*. The owners are those that would own the intellectual property within the system. Implementers would be those particular agents that are responsible for the building, implementation, deployment and maintenance of the system. These can be contractors that work directly for the owners as well as individual employees.
- *Operators*: Agents that interact directly with the system. They provide the system with inputs and receive the system's outputs.
- *Executors*: These are the agents who make decisions that are informed by the system. Executors receive information from operators.
- *Decision-subjects*: Agents who are affected by decisions made by the executor(s).

- *Data-subjects*: Agents whose personal data has been as input or to train the system.
- *Examiners*: Agents auditing or investigating the machine learning system. Depending on the system, they may interact with one or more of the other roles and the machine learning system itself.

With regards to explanation, there are several kinds of demands an agent might have. Barredo Arrieta et al. [2020] have comprised a list of the researched goals that might be achieved through model explainability. In line with these we propose the following goals:

- *Informativeness*: The ability to support decision-making by expressing information with proxies that are human-understandable.
- *Causality*: Provide intuition for finding (and potentially validating) cause-and-effect relationships, by revealing correlations between data variables.
- *Trustworthiness*: The capabilities of an explanation in inducing trust in the model (i.e. the confidence of whether a model will act as intended when facing a given problem).
- *Confidence*: The capability to convey information about the robustness, and stability of the model's working regime and underlying data.
- *Fairness*: The capacity to guarantee fairness and subsequently expose socially unjust discrimination.
- *Transferability*: Applying a model to new problems by making sure the model and its boundaries are fully understood.
- *Accessibility*: Allowing more end-users to get involved by increasing non-technical or non-expert information artefacts.
- *Interactivity*: the ability to facilitate interaction between model and user from an explanation interface.
- *Privacy awareness*: Creating insight into the complex patterns that a model has learned, can expose a breach of privacy. Contrarily, explainability can be detrimental to privacy when third party explanations compromise the differential privacy origin of data.

When we view all agents as separate stakeholders, we can explain the goals they might want from a model. It is important to note however that all stakeholders share a common goal, informativeness.

If you recall, we subdivided *creators* into implementers and owners. *Implementers*, such as developers and researchers, seek to ensure and improve product efficiency, as well as research and add new functionality. They design, build or facilitate the development of the AI models. They are primarily focused on explaining their models as a means for quality assurance, that is to aid system testing, debugging and evaluation as well as prove the robustness of a model. Thus, they seek out informativeness and confidence. *Owners*, such as product owners, managers and executive board members, want to understand corporate AI applications and assess regulatory compliance. They rely on the potential of a system to make money. Therefore they demand informativeness, accessibility and causality.

The *examiners*, among whom are the auditor and policy-makers, need to certify model compliance with the legislation in force and carry out audits in general. In that way, they want informativeness, privacy awareness, fairness, confidence and causality.

The *operators*, i.e. all users that directly interact with the system and its parameters, need the system to be easy to use, in order to analyse data and make knowledge available. Thus, they want informativeness, interactivity and accessibility.

The *executors*, e.g. domain experts and advisers, want to trust the model itself and be informed by the given knowledge. So they want informativeness, causality, trustworthiness and confidence.

The *decision-subjects*, usually end-clients or consumers, want to understand their situation and verify fair decisions. In other words, they want informativeness, trustworthiness, fairness, accessibility, interactivity. It is important to note that consumers often have little to no practical knowledge of the topic at hand.

The *data-subjects*, which depends on the dataset (but usually also consumers), want to be assured that their data is safe. Thus, they need privacy awareness.

In a practical environment, stakeholders are often part of several agent groups at once. For example, domain experts are often both operators and executors at the same time.

Furthermore, in some situations, AI-driven applications are completely geared towards consumers and, thus, have an interface that the consumer interacts with directly. Within these consumer-oriented models, Consumers are usually operators, executors as well as decision-subjects, and data-subjects, as they deliver inputs for the system. For example, this is the case with the model explained by company E. In such situations, consumers will usually only interact with the explanations, if present, and not with the rest of the model's parameters. An example of this situation is also covered in the third scenario.

2.2 Development of methods for explainability

In recent years, several ante-hoc and post-hoc XAI-methods have been developed to increase model transparency and explainability. Vilone and Longo [2020] have compiled a list of all current ante-hoc and post-hoc XAI-methods. Figure 3 depicts the type classification of XAI-method as branches, in accordance with the characteristics described in Subsection 2.1.3 and shows the distribution of the methods across these branches. A full list of these methods can be found in Vilone and Longo [2020] (grouped by their respective applicability), but to give an understanding of the capabilities of such methods we will give a few examples, inside and outside the insurance industry.

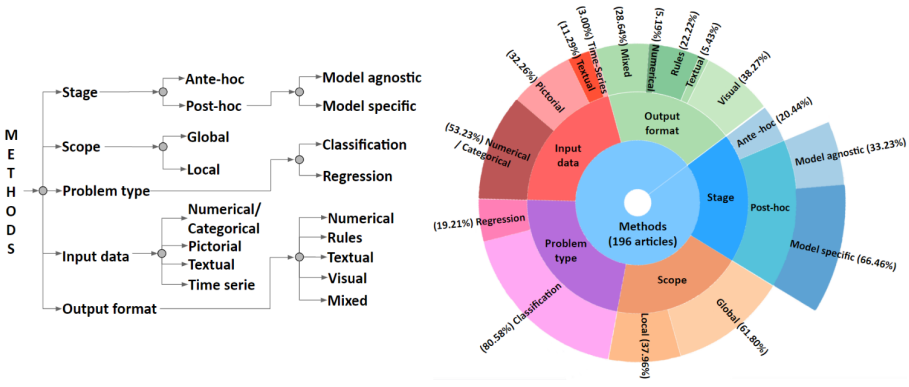

Figure 3: Classification of different methods for explainability
[Vilone and Longo, 2020]

One of the simpler and more commonly known examples of XAI-techniques is Local Interpretable Model-Agnostic Explanations (LIME) [Ribeiro et al., 2016]. As the name suggests LIME is a model-agnostic, post-hoc technique and can produce several different output types. It can only give information about a single instance at a time (e.g. one image at a time), meaning its locally faithful. A good example of its capabilities is classic wolf-husky image classification example. Ribeiro et al. [2016] tasked a logistic regression classifier with classifying 60 photos of wolves and Eskimo Dogs (huskies). They used a training set of 20 hand-selected images. Note, that the classifier had an accuracy score of 97%. They also used LIME to highlight those particular pixels in every image that contributed most to the classified outcome. Figure 4 depicts six of the classified instances, including one of the two, incorrectly labelled instances. As is visible in the LIME predictions, the model classified an image to be of the category wolf, based on the presence of snow in the background (or a light background at the bottom), and of the category husky, based on the absence of snow, regardless of animal colour, facial features, position, pose, etc. This highlights an undesired correlation used by the classifier to determine the outcome (more on this phenomenon in Section 2.3.1).

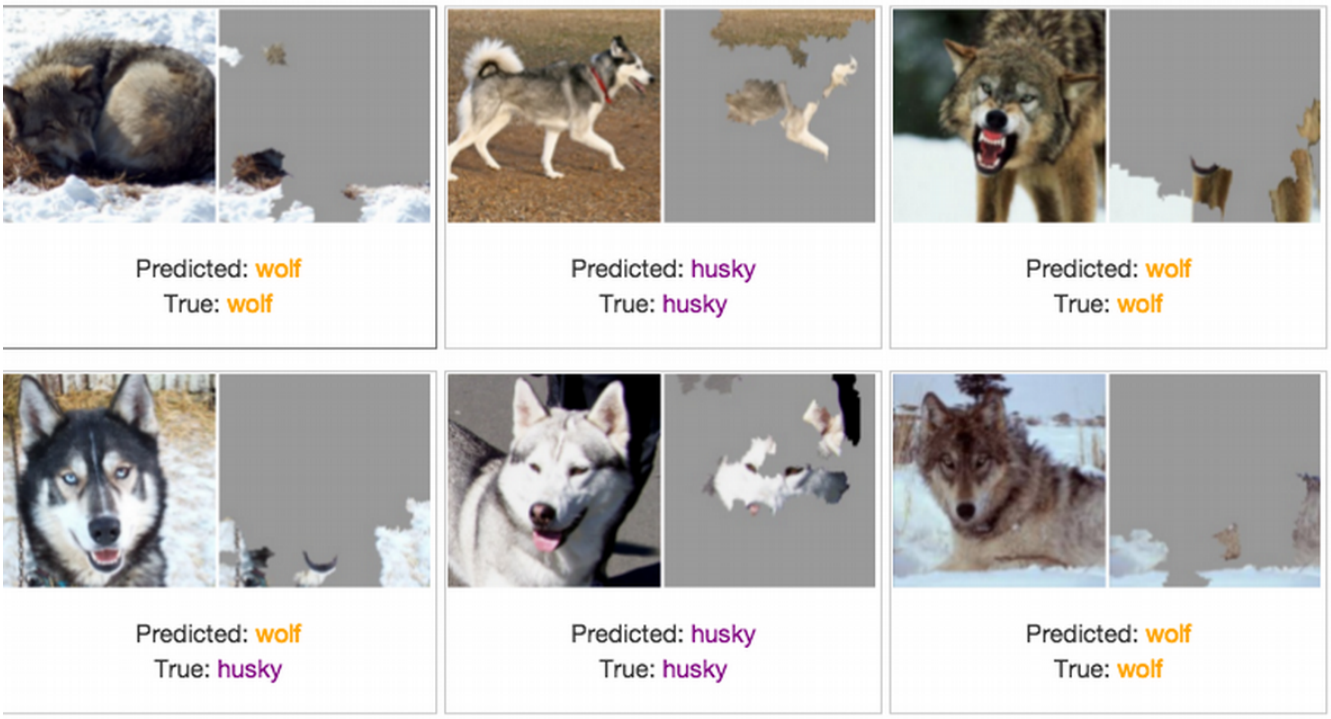

Figure 4: Raw data and explanation of a bad model's predictions in the "Husky vs Wolf" task [Ribeiro et al., 2016]

Interestingly, Ribeiro et al. [2016] developed this bad classifier intentionally to show the effects it has on user understanding, hand-picking those particular images for the training set, where all pictures of wolves had snow in the background, while pictures of huskies did not. They first showed graduate students (who had taken at least one ML course) 10 classified pictures without LIME explanations, including the two incorrectly classified images (where one wolf is not in a snowy background and is thus predicted 'Husky', and one husky is and is thus predicted as 'Wolf'). They then asked each subject three questions: 1. "Do you trust this algorithm to work well in the real world?", 2. "Why do or don't you?" and 3. "How do you

think the algorithm is able to distinguish between these photos of wolves and huskies?". After collecting the responses, they showed the images with their associated LIME explanations, and asked them the same questions. Table 1, below, shows the observations they made.

	before	after
Trusted the bad model	10 out of 27	3 out of 27
Mentioned snow as a potential feature	12 out of 27	25 out of 27

Table 1: 'Husky vs Wolf' experiment results [Ribeiro et al., 2016]

This experiment demonstrates the utility of explaining individual predictions to gain insights into classifiers, knowing when (not) to trust them and why.

Similarly, LIME can be used for many other problems and algorithms. Hofman [2018] describes in a case study, how Zelros uses LIME to explain their neural network that predicts insurance claim complexity. Figure 5 shows LIME's output. This time in the form of feature importance (every attribute's negative or positive contribution to the classified outcome.

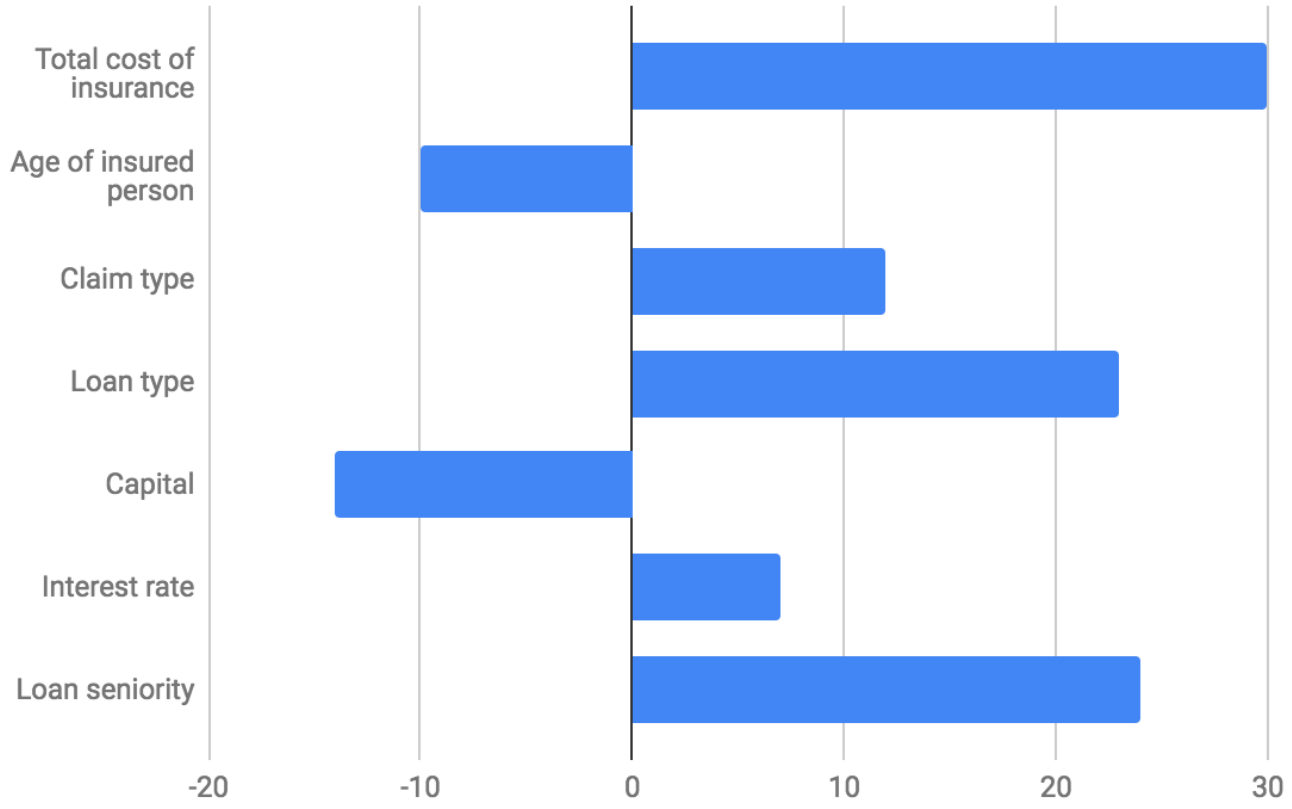

Figure 5: LIME Features importance of Zelros' neural network based insurance claim complexity prediction system [Hofman, 2018]

Hofman [2018] also used Shapley Additive Explanations (SHAP) [Lundberg and Lee, 2017], a similar more recently developed method model-agnostic method that builds on LIME and other methods, in a similar context. SHAP uses game theory (Shapley values, specifically) as a way to reverse-engineer the output of any predictive algorithm. Figure 6 shows a SHAP summary plot, which combines feature importance with feature effects. Each point on the summary plot is a Shapley value for a feature and an instance. The position on the y-axis is determined by the feature and on the x-axis by the Shapley value. The colour represents the value of the feature from low to high. Overlapping points are distributed in y-axis direction, so we get a sense of the distribution of the Shapley values per feature. The features are ordered according to their importance.

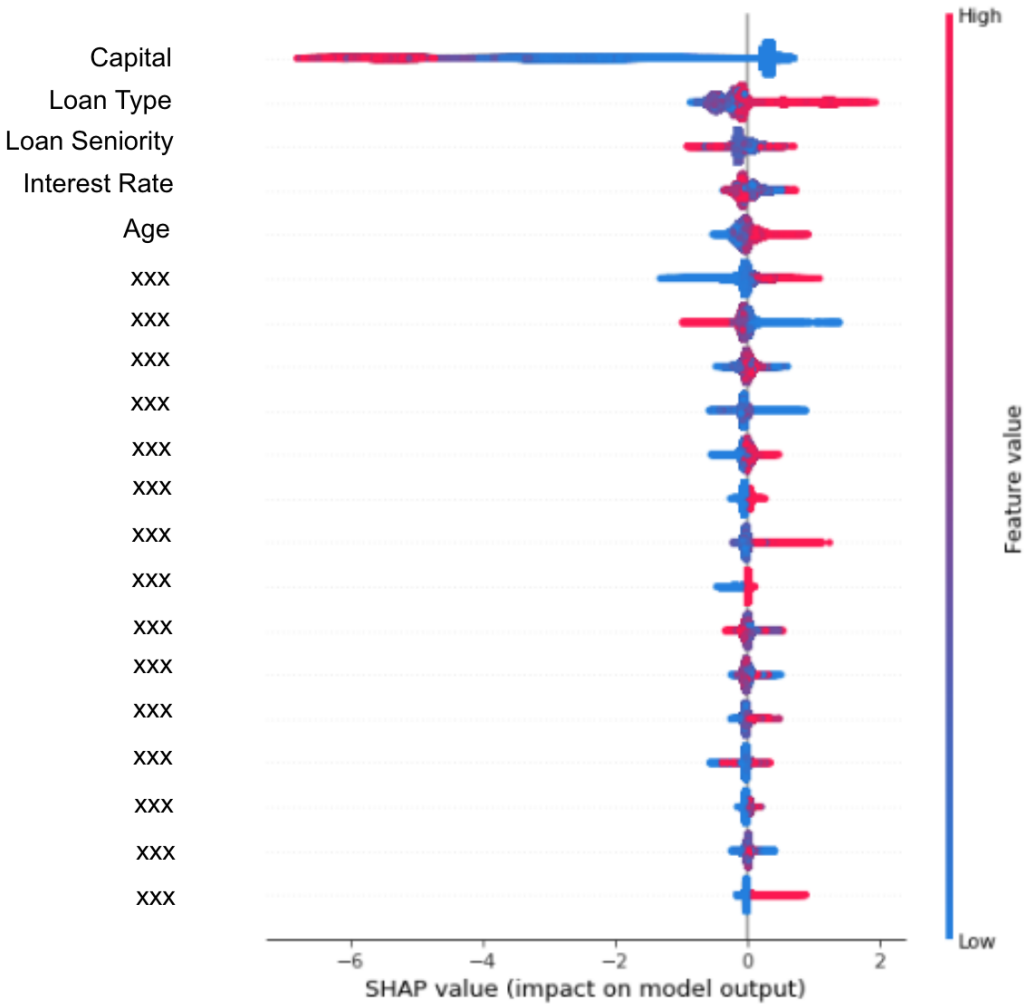

Figure 6: SHAP summary plot of Zelros' predictive insurance claim management system [Hofman, 2018]

Granted, this is only the tip of the proverbial iceberg. Other methods have been developed to address a wide array of different explainability related problems. Many of which are listed by Vilone and Longo [2020].

2.3 Responsible AI

2.3.1 Bias

AI bias is a phenomenon that occurs when an algorithm produces results that are systemically prejudiced due to either erroneous (or correct but undesired) assumptions in the machine learning process (as is the case in our example of the wolf-husky classifier in Section 2.2). This can happen with learning algorithms when they are trained on their dataset. In this case, we call the phenomenon *algorithmic bias*. Even so, other forms of bias exist. It can also result from human errors (e.g. faulty collection or representation of input data). Rule-based systems can also be biased when rules are formed incorrectly by the domain experts.

According to Barocas and Selbst [2016], bias can always be linked to one of the following sources:

- *Skewed data*: Bias within the data acquisition process.
- *Tainted data*: Errors in the data modelling definition, wrong feature labelling, and other possible causes.
- *Limited features*: Using too few features could lead to an inference of false feature relationships that can lead to bias.
- *Sample size disparities*: When using sensitive features, disparities be- tween different subgroups can induce bias.
- *Proxy features*: There may be correlated features with sensitive ones that can induce bias even when the sensitive features are not present in the dataset.

2.3.2 Ethics & fairness

Sometimes biases can be functionally correct, but immoral to base results upon. These undesired biases stem from a black-box model's tendency to unintentionally create unfair decisions by including sensitive factors such as the individual's race or gender. This phenomenon gives rise to certain discriminatory issues, either explicitly (considering sensitive attributes) or implicitly (considering factors that correlate with sensitive attributes). This implicit bias can be especially hard to identify. For example, credit scores are derived from objective data that exclude race as an attribute, and are highly predictive of insurance losses. Thus, most regulators allow the use of credit-based insurance scores. However, credit scores are also highly predictive of skin colour in some areas of the US, acting in effect as a proxy for race. For this reason, California, Massachusetts, and Maryland don't allow insurance pricing based on credit scores [Schreiber, 2019]. According to Schreiber [2019], in a 'perfect model', which differentiates not based on race, gender, or religion alone, but rather, takes all individual factors into account to give a personalised score for a single person, the adverse discriminatory effects of these biases are forgone, giving a perfect score which ultimately assesses only your individual risk. Unfortunately, such models do not exist yet.

Discussions concerning algorithmic bias first sparked about two decades ago. However, in recent years this debate has taken a practical turn. There is a growing awareness of this algorithmic bias and a need for responsibly built AI. New ethical expert roles are being appointed in companies and there is an increase in efforts taken against bias across domains [Rakova et al., 2020]. Transparency and explainability are the latest tools in combating this problem. Post-hoc methods especially can help expose the biases and

causalities that AI systems are based on, by analysing how output behaves with respect to the input features. It is then down to the implementers and users to choose which biases would be deemed unfavourable in the context of their domain.

Barredo Arrieta et al. [2020] propose three main categories of fairness that should be considered. *Individual fairness* which is analysed by modelling differences between one subject and the rest of the population, *group fairness* which regards fairness from the perspective of all individuals, and *counterfactual fairness* which tries to interpret causes of bias. With all this in mind, according to ethically acceptable trade-offs can be considered, should they be reasoned, explicitly acknowledged, well documented, evaluated. The decision-maker should then be held accountable for making the appropriate trade-off, prior to development. In this case, redress must be possible as a mechanism to guarantee that unjust adverse impacts are attended to.

The 'High-Level Expert Group on AI presented Ethics Guidelines for Trustworthy Artificial Intelligence' [Smuha, 2019], drawing on the Charter of Fundamental Rights of the EU and international human rights law, prescribes four ethics principles in the context of AI. Namely, *respect for human autonomy*, *prevention of harm*, general *fairness* and *explicability*. Smuha [2019] also cites 7 key requirements:

1. *Human agency and oversight*: Fundamental rights, human agency and human oversight.
2. *Technical robustness and safety*: Resilience to attack and security, fall back plan and general safety, accuracy, reliability and reproducibility.
3. *Privacy and data governance*: Respect for privacy, quality and integrity of data, and access to data.
4. *Transparency*: Traceability, explainability and communication.
5. *Diversity, non-discrimination and fairness*: Avoidance of unfair bias, accessibility and universal design, and stakeholder participation.
6. *Societal and environmental well-being*: Sustainability and environmental friendliness, social impact, society and democracy.
7. *Accountability*: Auditability, minimisation and reporting of negative impact, trade-offs and redress.

Furthermore, regarding fairness jurisdiction in the EU, Article 21 of the Charter of Fundamental Rights of the European Union states that "any discrimination based on any ground such as sex, race, colour, ethnic or social origin, genetic features, language, religion or belief, political or any other opinion, membership of a national minority, property, birth, disability, age or sexual orientation shall be prohibited." [European Union, 2010].

Ruf et al. [2020] propose several approaches in contributing to fair AI, which include:

- Removing sensitive attributes from the dataset (i.e. fairness through data sanitation).
- Treating similar individuals equally based on an adequate distance metric.
- Minimising the absolute difference of outcome distributions of all groups.
- Optimising towards equal positive and negative classification rates across all groups.

Finally, according to Barredo Arrieta et al. [2020], AI products should be in line with the United Nation's Sustainable Development Goals [United Nations, 2015] and contribute to them positively and tangibly, benefiting humanity and the common good.

3 Methodology

In this chapter we define the methodologies used to design and evaluate our checklist.

3.1 Design science research

Contrary to behavioural science, design science is focused on designing an artefact. In our case, this artefact is a checklist for explainable and transparent AI-applications. The checklist design will be discussed in greater detail in Section 5.

3.1.1 Design cycles

Design science research is characterised by three iterative cycles shown in figure 7. The relevance cycle, the rigor cycle and the design cycle. In the relevance cycle, requirements are set in by studying the problem and risks from the contextual environment. These requirements are refined further via field testing. Thus, we do four exploratory interviews with several major players in the dutch insurance industry to gain perspective on the current adoption of AI-algorithms as well as corresponding ex-ante and post-hoc techniques. Furthermore, we test the usefulness of our checklist with confirmatory interviews In the rigor cycle, academic literature and similar pre-existing artefacts are consulted to form a knowledge base to draw theories and methods from. The findings of the exploratory interviews will also be added to this knowledge base in order to gain expert experience and perspective. The requirements and insight from the knowledge base will be considered in every design cycle. In this internal cycle, the artefact is built and evaluated. That is to say, the artefact will be assessed conforming to the set of requirements and if it is still in line with the existing theories and methods.

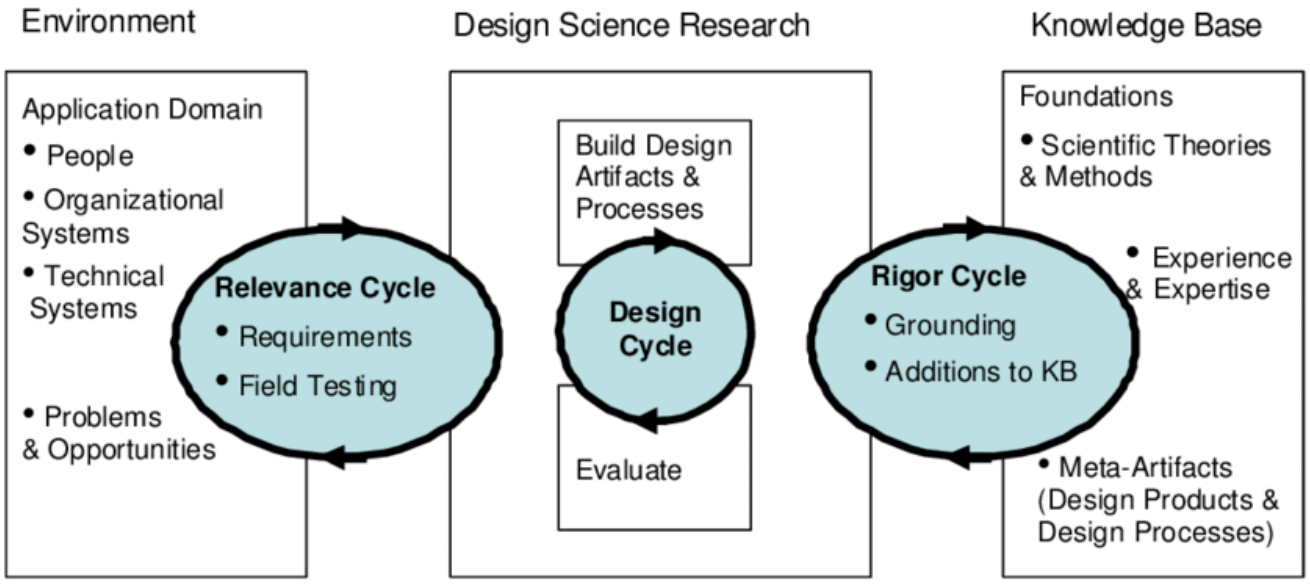

Figure 7: Design Research Cycles
[Hevner, 2007]

3.1.2 Design Guidelines

During the design of our artefact, we keep in mind the seven guidelines from the design research paradigm for information systems [Hevner et al., 2004]. Although these guidelines are not to be followed strictly, they will serve as support in making sure we consider all facets of the design process. The guidelines, formulated identically as those listed in Hevner et al. [2004], are the following:

1. **Design an artefact**: Design-science research must produce a viable artefact in the form of a construct, a model, a method, or an instantiation.
2. **Problem Relevance**: The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
3. **Design Evaluation**: The utility, quality, and efficacy of a design artefact must be rigorously demonstrated via well-executed evaluation methods.
4. **Research Contributions**: Effective design-science research must provide clear and verifiable contributions in the areas of the design artefact, design foundations, and/or design methodologies.
5. **Research Rigor**: Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artefact.
6. **Design as a Search Process**: The search for an effective artefact requires utilising available means to reach desired ends while satisfying laws in the problem environment.
7. **Communication of Research**: Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

3.2 Interviews

We conduct two types of interviews during our research. Firstly, we will perform semi-structured exploratory interviews to assess the current state of the art with respect to AI-techniques used within the insurance industry. This is helpful because, while current literature tells us a lot about the possibilities and practices of AI, it gives little insight into the actual adoption of AI-techniques. Additionally, practical context is usually missing in most available literature. We will interview four industry experts at companies that operate within the insurance industry, ranging from software developers to outright insurance companies. The findings can be found in Section 4.1.

Secondly, we conduct confirmatory interviews with similar companies to evaluate our findings and checklist design. This will give us a strong indication of both the correctness, robustness as well as practicality of our design. similarly to the exploratory interviews, evaluation is done by conducting four confirmatory interviews with industry experts from insurance companies, and their software suppliers. The evaluation findings can be found in Section 6.1.

For privacy reasons we will not disclose the actual company names, nor will we disclose employee names and information that could be used to identify firms and personnel. As such, we will call the seven companies ‘company A’ through ‘company G’, respectively. Note, that we have conducted both an exploratory interview as well as a confirmatory interview with company B.

4 Exploratory interviews

In this chapter, we cover the results and findings of the exploratory interviews. As mentioned, we carried out four exploratory interviews with industry experts from financial institutes, insurance companies, and their software suppliers. For privacy reasons we will call these companies: 'company A', 'company B', 'company C' and 'company D', respectively. A short summary of these companies is shown in table 2, below.

	company A	company B*	company C	company D
core business	insurance software supplier	insurance intermediary (software)	financial services	financial services
# of employees	100+	100+	1.000+	10.000+
job-title interviewee(s)	product owner	ML-engineer & software engineer	manager client contact financial services	Product Owner & Innovation Manager Business Automation
uses RBS for	various processes	various processes	various processes	various processes
uses ML for	-	fraud detection	-	fraud detection
uses DL for	-	-	-	-

Table 2: Summary of exploratory interview company information
*Also takes part in confirmatory interview

The focus of these interviews has three primary goals in mind. firstly, we want a more in-depth understanding of the insurance industry and its related processes, stakeholders, demands and concerns. Secondly, we want to know which AI-techniques said companies deployed and which they plan to deploy in the future. Thirdly, we want to know how the industry values transparency and explainability. Our most notable findings are explained below.

4.1 Findings

The insurance industry is still in a preliminary phase when it comes to the deployment of AI-technologies. All of the four companies interviewed have deployed some form of Rule-Based System, but some are hesitant to adopt more complex AI-techniques (like Machine Learning and Deep Learning). This is because most companies are focused on improving these Rule-Based Systems and ironing out any inefficiencies. Additionally, the interviewees from company B expressed a growing concern that people lack trust in Machine Learning algorithms. It takes time and effort to convince people that Machine Learning algorithms work better than then Rule-Based Systems, even though oftentimes they are statistically proven to do so. Thus, at the moment, all of them prefer Rule-Based Systems because they are more explainable, even

when the results that are explained are sometimes less accurate.

Two out of four companies interviewed, have already deployed ML algorithms in their processes. Notably, both company B and D have deployed the technology for insurance claim handling, specifically to detect possible fraudulent activity. It is interesting to note however, that both companies do so with different kinds of algorithms. Company B uses a decision tree classifier (supervised learning) algorithms, whereas company D uses a k-means clustering algorithm for anomaly detection (unsupervised learners). Company B also uses ML algorithms for calculating car insurance premiums. They are also experimenting with ML algorithms to calculate a customer's risk coverage ratio, but this system has yet to be deployed. Importantly neither use non-transparent algorithms (including Deep Learning) for these tasks.

Company B and D are already experimenting with Deep Learning. However, none of the interviewed companies have deployed any Deep Learning systems thus far. There are four main reasons for this:

1. **DL is less explainable**: Even though Deep Learning algorithms are usually more accurate than Machine Learning algorithms (and Rule-bases Systems), they are even less explainable. A balance between accuracy and explainability has to be found. Insurance companies mostly choose explainability in favour of a marginal increase in accuracy. They need increased explainability to understand and convey why the system gives a certain output, otherwise, the results are not actionable. To give an example, if an insurance claim is given a certain 'anomaly score', the system can mark the claim as potentially fraudulent. In such a case a claims handler has to evaluate the claim information to look for peculiar activity. If none can be found by the claims handler, no further action can be taken, as a claim cannot be marked as fraudulent based on its 'anomaly score' alone. In other words, the insurance company cannot take action when the results come from an insufficiently explainable system. Until the moment the system itself can convey why it deems these claims as potentially fraudulent, the system can only give an indication of which claims are potentially fraudulent and nothing more.
2. **Understanding DL requires technical/mathematical expertise**: Because Deep Learning algorithms (and some Machine Learning algorithms for that matter) have such high complexity, they are less explainable. Thus, they require more expertise to be understood and used (this is the case for all system agents roles, but especially operators and executors). For most current employees this creates a knowledge gap that is hard to overcome. Furthermore, if they were to overcome this obstacle, their job description would change significantly. Claims handlers would turn into model experts.
3. **DL is less transferable**: Deep Learning algorithms are sometimes less transferable than other AI-techniques. The input data that insurance companies use at the moment is less suitable for these types of algorithms. Additionally, some of the data that could be used to extract the most out of DL-algorithms are not present in the dataset or are off-limits due to privacy concerns.
4. **Streamlining RBS has more value in the short term**: More quality gain can be found in streamlining current Rule-Based Systems and ML-processes, instead of looking for accuracy gain with DL-algorithms. Gains can especially be made in the refinement of input data (e.g. feature selection), as this is where most resource and thought is going at the moment.

Future prospects for all companies range from the initial deployment of Machine Learning to the deployment of complex Deep Learning algorithms when the aforementioned issues start to be resolved. Most expect to start incorporating (more of) these complex AI-technologies near the end of the next five years.

4.2 What does this mean for our design?

The checklist should take into account that the insurance industry is still in a preliminary phase: Most companies use Rule-Based Systems exclusively. Some use Machine Learning but have only recently started to do so. These systems do not have agency, as actions are still taken by employees who use ML output as indications. Also, companies are still experimenting with said ML algorithms for other applications. Not a single company out of the four that we interviewed has deployed Deep Learning, as these algorithms are not transparent enough. Given this fact, our checklist must be relevant for the companies that only use Rule-Based Systems, but must also be a guide to ML- and DL-technology, in order to aid with applications in the future.

5 Checklist design

In this chapter, we explain the design process of the aforementioned checklist globally. We explain what the checklist should be, its purpose, constraints, the building phase and the design itself. Afterwards, we discuss its evaluation.

5.1 Purpose

The checklist should be a list of 'checks' that, if answered properly, should test the explainability and transparency of AI-model applications, as well as highlight potential weaknesses and areas for improvement.

A *check* is a component that features one or more questions, hence the collective is called a checklist. Every check comes with an elucidation to clear any confusion for the reader and to make sure the question is answered as intended. Checks either have open answers, multiple choice answers or both. To give an example of what a particular component would look like we will cover two components in detail in Section 5.4. The complete checklist can be found in Appendix A The checklist is designed with two main purposes in mind.

Firstly, it should be used to confirm quality and completeness of an AI-application with regards to its explainability and transparency. In that way, the checklist can essentially be used as a guide to evaluate if all facets, that make a well designed explainable and transparent AI-application, are accounted for. If, based on this checklist, one would conclude their application is not complete or lacks quality in certain key areas, it is hoped that this would serve as an indication where further progress should be made.

Secondly, the checklist, if properly filled in, could be shared with third parties (clients or companies) to show the quality and completeness of their AI-applications with regards to explainability and transparency. This is especially helpful for collaboration between companies to give confidence that certain information or assets can be shared. Moreover, this could be interesting from a marketing standpoint, giving clients assurance that your application is well designed and responsible.

5.2 Constraints

To fulfil these purposes that checklist design has to meet a certain set of requirements, similar to the design guidelines proposed in Section 3.1.2. This is way we propose several constraints. The checklist is based on an existing checklist[1] named 'Checklist-KOAT' or 'Checklist Kwaliteit Onbemenste Advies- en Transactietoepassingen' by SIVI, a Knowledge and standardisation institute for financial services. This existing checklist covers several topics with regards to unencumbered computer applications for financial advice and financial transactions. We can deduce several helpful constraints that are implied in this checklist. We will use these implied constraints as well as our design guidelines to set constraints to design our checklist. The following constraints are used set:

1. **Practical Relevance**: We want our checklist to be applicable for practical use. That means that all covered topics should be relevant from a practical standpoint. Furthermore, The checklist cannot be overly long or be too technically in-depth, as this would disincline people from using it.

[1]The Checklist-KOAT can be found at https://www.sivi.org/checklist-koat/

2. **Non-expert terminology**: The checks and elucidation should refrain from using expert terminology as much as possible. If used in a practical environment by actual employees of financial companies, expert terminology may be unclear and would not induce a full understanding of the topic being covered.
3. **broadness-precision balance**: Topics should be covered broadly enough to be appropriate for most, if not all, AI-model applications. Yet, checks should be precise enough to get the most informative answer. A proper balance should be found between these two ends.

The checklist designed should adhere to all mentioned constraints as much as possible.

5.3 Building phase

With the knowledge gained from our literature review and exploratory interviews (sections 2 and 4 respectively), as well as with the aforementioned purposes (Section 5.1) and constraints (Section 5.2) in mind, we designed an initial draft of the checklist. From there design was done in an iterative fashion. The initial draft was evaluated through a confirmatory interview with company E. The design was further improved to create a second draft, which was evaluated through a confirmatory interview with company F. This process was repeated until all four companies where interviewed and a final draft was conceived. The full checklist design can be found in Appendix A. We will cover the subjects and two examples in detail in Section 5.4, below.

5.4 The designed checklist

As discussed earlier, the checklist should cover all relevant facets that result in a responsible AI-application. The literature review, as found in the Section 2, gives insight into all possible facets that could be taken into account. The exploratory interviews give context into the particular facets that are practically relevant for the purposes of our checklist. Thus, the designed checklist has the following components:

1. *Transparency of algorithm*: Deals with the inherent transparency of the underlying algorithm.
2. *Purpose*: Deals with the purpose that explainability is mainly used to fulfil.
3. *Development*: Deals with the way accuracy and explainability are prioritised in the development of the application.
4. *Impact*: Deals with the impact the results of the application have on the end-customer and what this is influenced by.
5. *Explainability by Design*: Deals with the Ante-hoc techniques used and how they increase explainability.
6. *Add-on explainability*: Deals with the Post-hoc techniques used and how they increase explainability.
7. *Explanation output type*: Deals with the type of explanation that is outputted.
8. *Stakeholders*: Deals with the relevant stakeholders and their demands.
9. *Redress*: Deals with the way end-customers can receive additional information when they have a question or a complaint.

10. *Bias*: Deals with possible undesired biases and how they are prevented.
11. *Expertise*: Deals with new expertise that is brought into the company to support the quality of the application.

The checklist design is structured according to the aforementioned constraints and follows the design research guidelines. Below is an example of what a check looks like.

Example 1:

#	Subject	Check	Answer	Elucidation
1	Transparency of algorithm	Elaborate whether the application itself is already transparent to the user or whether external techniques are needed to increase transparency and explainability?	Open answer	We can divide AI algorithms into two categories. Namely, *transparent algorithms* and *non-transparent algorithms*. Transparent algorithms include *rule-based systems*, *linear/logistic regression*, *decision trees*, *k-nearest neighbours*, *rule-based learners*, *general additive models*, *bayesian models*. Non-transparent algorithms include *tree ensembles*, *support vector machines* and *neural networks*. It is also possible that your application falls somewhere between the two extremes.

As you can see the subject, check, and its elucidation are given. There is an answer box where answers can be filled in, although for the purposes of this example, we have left it very small. An open answer is supposed to be given, as extensively and fully as possible.

The check has one main purpose. We would like to know how transparent a given application's underlying algorithm is in the first place. The answer to this question immediately gives an indication of how relevant the rest of the checklist is going to be. Thus, it is the first subject we ask about. The check complies with all the aforementioned constraints.

The following topics where excluded from the checklist: *Level of transparency* and *requirements of explainability*. They were excluded for the same reason. They were both too theoretical in concept and terminology, while their answers would ultimately not yield enough useful information. The full checklist can be viewed in Appendix A.

6 Evaluation

In this chapter, we discuss the evaluation of the designed checklist, what consequences followed from the chosen design philosophies, which obstacles presented themselves, and which obstacles were ultimately surmounted.

6.1 Confirmatory interviews

As mentioned in 3.2, similarly to the exploratory interviews, evaluation is done by conducting four confirmatory interviews with industry experts from insurance companies, and their software suppliers. For privacy reasons, we will call these companies: company B, company E, company F and company G, respectively. A summary of these companies is shown in table 3, below.

	company E	company F	company G	company B*
core business	software supplier	insurance & pensions	software supplier (insurance & pensions)	insurance intermediary (software)
# of employees	10+	1.000+	10+	100+
job-title interviewee(s)	CCO\CMO	sr. IT Architect	user interaction designer	ML-engineer & software engineer
use RBS for	various processes	various processes	various processes	various processes
use ML for	-	-	policy recommendation	fraud detection
use DL for	-	-	-	-

Table 3: Summary of confirmatory interview company information
*Also takes part in exploratory interview

Other than to evaluate our design, these interviews essentially helped us confirm whether our original findings are correct, and if they still hold within new contexts. The structure of the interview is as follows: For every check and its elucidation, we ask three things: 1. Is the phrasing and meaning clear? 2. How relevant is the check and its encompassing subject (with regards to the purposes mentioned in 5.1)? and 3. What would your answer be to the question for your specific application? After all subjects are covered, we ask two general questions about the entire set of subjects: 1. Do you deem the sequence/order of subjects logical and favourable? 2. Is the set of subjects (and checks) complete or do you think a subject is missing?

As noted earlier, conforming to our used design science research methodology, evaluation is done during the design phase. Therefore, the design process has an iterative nature. Consequently, a new checklist draft is designed after each confirmatory interview. This way the design is improved in a step by step manner. Because we do not openly discuss the transcripts in this thesis (due to the aforementioned privacy reasons),

we think it is wise not to give examples of which specific remarks gave rise to which specific changes in the design. Rather, we will present an indication of how the interview process and its results affected parts of our design. Most initial constructive criticism, in the interview with company E, was aimed at phrasing and meaning (of checks and their elucidation) being unclear. This resulted in the inclusion of extra illustration were needed, or rephrasing of said unclear pieces of information. This was the case throughout the design. In most cases, an illustrative example was also added in an attempt to clear up any remaining confusion. The next iteration was found to be much more clear and comprehensible, although slight improvements kept being made from version to version. Until, in the last interview, no confusion was remarked explicitly.

The initial design featured several questions which delved into the explainable requirements (mentioned in 2.1.2) and the levels of transparency (first mentioned in 2.1.1), with separate checks for each subsequent requirement and level of transparency. After the interview with company E, these checks were removed due to their overly in-depth and theoretical nature, while seeming to yield little actionable result. In contrast, two new topics were added, which were deemed to be missing, despite being rather important in hindsight. These where the checks from the topics; 'transparency of algorithm' and 'purpose', which can be found in the final design. The latter's answer type was changed from multiple-choice to open, as several interviewees remarked that a model's transparency could be considered fluid, rather than binary (transparent or opaque). This also made it easier to categorise those AI-models that we did not include in Section 2.1.4 and 2.1.5 and, subsequently, our elucidation. This a clear example of our initial knowledge being tested in a practical environment.

In the first draft, the check for the topic 'stakeholders', was phrased: "For each relevant stakeholder, explain 1) what demands they have in regards to explainability of the application, and 2) how these demands are fulfilled by said explainability". This means the form-filler would have to decide for himself which stakeholders existed, why and which were relevant to explainability (and which demands they might have). After the interview with company E, this was changed, as this was too much to ask from the form-filler. Several stakeholders (from a job-description perspective) were now explicitly listed. In the next iteration, after the interview with company F, a set of 4 demands (which is a simplified set of the demands listed in Section 2.1.6) was added to help the form-filler pick the right ones. The interviewee from company G, expressed that this check would be most effective if all demands from Section 2.1.6 were listed, while presenting the stakeholders (from a system agent perspective) as mentioned in 2.1.6. The interviewees from company B shared this sentiment.

6.2 Discussion

In the final iteration of the design, all checks and elucidations seemed to be phrased clearly, to be fully understood by the interviewees, based on our assessment of their answers. Also, based off the results, all checks and topics present in the final iteration seemed to be relevant enough to be included in the design. Interviewees specifically expressed relevance for the topics spanning bias. Given this fact, more checks could be added towards this topic topic. Such questions could dive deeper into why they include and exclude certain biases in their model (thus, revealing which biases they would label as undesired biases). This particular check comes to mind because the interviewees from company B, who have put lot of thought into bias, expressed doubt that any form-filler would choose to explicitly reveal such debatable and delicate information. As they feel that, although some biases are mentioned without hesitation, some biases fall more towards the controversial side of the spectrum. Even so, we hesitate to make the checklist

even longer, because this would discourage its ample and extensive completion. Also, we feel that our intention is not to pressure form-fillers to expose their reasoning and concerns about the biases that they choose to include or exclude, rather we want them to actively think about biases and their implications, and hopefully engage in a conversation (either internally or externally).

Furthermore, in our literature review, we did not explicitly cover natural language processing (NLP) as a distinct field-of-interest within the AI domain, as not much literature covers explainability in this field. As such, we only mentioned the topic very briefly in Section 2.1.5. However, much to our relief, when interviewing company F about their NLP technology, all checks seemed to be just as relevant and accurate for NLP. This further affirms our belief that our checklist design is applicable for all AI-fields (at least having proven its use for all in-interviewed discussed applications).

Eventually, we landed on a design that puts a heavy emphasis on questions formulated with open-ended answer in mind. This has two main advantages, whilst also running the risk of some potential drawbacks. The first advantage is that phrasing the questions in such a way, tends to squeeze as much interesting information out of a single check as possible, as long as the interviewee is motivated to explore the answer to the intended extent (interviewees have at least expressed the intention to do so). The second advantage is that this open-ended phrasing creates room for a certain broadness in the scope of a check's applicability (as mentioned in 5.2). By restricting the answers too much, you run the risk of excluding some AI-applications, rendering the check useless for their specific model.

These two advantages stand, provided that the form-filler completely understands the question asked and fully comprehends the question's intentions. Otherwise, if the intention of the question is lost on the form-filler, they cannot give a satisfactory answer. This risk could be amplified by the 'open-endedness', potentially leaving too much room for interpretation. During our interview with company E, the interviewee expressed the following concern: "I would like to encourage you to make checks as quantifiable, or in other words, objectively measurable, as possible. This way, you avoid simply entering "yes" everywhere and as a way of just ticking off boxes." Although we agree with this sentiment, we stress that the checks need to be broad enough to encompass all relevant AI-technologies, as to not exclude any application types. Even so, we think that, by choosing open-endedness over strict demarcation, the form-filler is forced to explain their reasoning about a subject, defeating the option to just fill in "yes". Also, we feel that too much hand-holding (by spelling out the possible answers to a question, or implying the possible answers in the question) usually leads to uninteresting and plain answers that yield little to no useful or actionable information. Luckily, as mentioned earlier, the interviewees expressed a full understanding of the checks and their elucidation. It is important to keep in mind, however, that each interviewee possessed a fair amount of expertise in the field. Because of this, we should explicitly remark that the checklist is best suited for similar individuals.

At the end of the day, as we discussed in Subsection 4.2, the checklist needs to be relevant for those companies that only use Rule-Based Systems, but must also be a guide to ML- and DL-technology, in order to aid with applications in the future. Based on the results of our confirmatory interviews, we feel our design fulfils this ambition.

7 Conclusions

In this thesis, we conducted the research and design of a checklist for insurance companies and their software suppliers, that should help assure quality standards of AI-applications and ensure a solid foundation for cooperation. First, we examined existing literature and conducted four exploratory interviews with industry experts to assess the current state of AI-adoption in the industry as well as uncover their issues and concerns. From here we set necessary constraints and started designing and evaluating the checklist in parallel. Evaluation was done through conducting four more confirmatory interviews with industry experts to assess if all prior knowledge had been applied correctly.

7.1 Answering the research questions

To design an effective checklist we needed to answer the main research question and the two subquestions proposed in 1.3. Our main research question reads:

"How can we assess the efforts taken to increase model transparency and increase its explainability in the organisational environment, making sure all practical and ethical facets are taken into consideration?"

Essentially, we want to know what influences model transparency and explainability, and how model transparency and explainability, in turn, affect the applications related processes and stakeholders in the organisational environment. Tackling this question gave rise to two subquestions.

7.1.1 "What are the factors that impact model transparency and explainability?"

Model transparency and explainability are influenced by various factors. We can divide these factors into two categories, namely the technological factors and theoretical factors. On the technological side we talked about how the underlying algorithm used makes a big impact. There are what we call inherently *transparent algorithms* and *non-transparent algorithms*. As talked about, transparent algorithms reach a certain level in all of the three categories of transparency, whereas non-transparent algorithms do not do well in either category. To become more transparent they would need external techniques that increase transparency and explainability. We've highlighted that these techniques come in two separate forms. Namely *ante-hoc*- and *post-hoc* techniques. *Ante-hoc* explanations are techniques that aim to increase explainability from the beginning to the end of all preparation stages (i.e. data cleaning, feature selection, or in the case of ML, training the model). *Post-hoc* explanations are those techniques that occur at run time or use the model's output itself. Moreover, *Post-hoc* techniques come in several different varieties, depending on their scope, transferability and output type. On the theoretical side, we find that there are specific goals that transparency and explainability are used to achieve, with different characteristics and types of explanations to achieve them. As mentioned, several general requirements can aid in this endeavour. All of these factors influence transparency and explainability, and consequently, how and by whom they should be used. Which brings us to our second sub-question.

7.1.2 "How do model transparency and explainability impact its related processes and stakeholders?"

In a practical environment, several stakeholders have demands for model transparency and explainability. We argue that it is best to view these stakeholders as agents in a confined AI-system. That way we

distinguish six agents, namely *creators*, *examiners*, *operators* (user), *executors* (user), *decision-subjects* and *data-subjects*. Their demands with respect to explanations include *informativeness*, *causality*, *trustworthiness*, *confidence*, *fairness*, *transferability*, *accessibility*, *interactivity* and *privacy awareness*.

It must also be recognised that algorithms which train on data specifically, can result in undesired or unfair biases. It is a moral obligation of the model's creators and operators to prevent and combat these biases as much as possible. Of course, where a line is drawn between informative rules and unwanted biases is a topic that is harder to pin down. In any case, model transparency and explainability can be a means to spot and prevent these biases where they might occur.

7.2 Contributions

In this chapter, we discuss the contributions made to the state-of-the-art. In our eyes, several contributions stand out when compared to other literature that we could find on the topics of XAI and AI in general.

Firstly, we propose a checklist that can be used to assess and help assure transparency and explainability for AI-application in a practical environment. It can also be used to verify enough thought has gone into the application and to share quality standards across parties.

Secondly, we give insight into AI and XAI adoption in the insurance industry. Few other papers talk about AI and XAI in the financial sector. Presumably, this comes down to the fact that most financial companies are only now starting to adopt AI-algorithms effectively, as knowledge on the subject has only started to grow in recent years. After all, once a technology has been discovered, it takes some time for it to develop into a commercially viable product. To verify this presumption we carried out exploratory interviews. These interviews confirm the hypotheses that the Dutch insurance industry is still in its preliminary phase when it comes to Machine and Deep Learning. Consequently, a big emphasis so far has been given to Rule-Based Systems, as they are relatively practical, simple and have been around for a little longer. They are also generally readily explainable when compared to ML- and DL-algorithms. Only a small number of the big Dutch insurance companies have already incorporated ML algorithms into their products. Not a single company interviewed had already applied Deep Learning in any practical fashion, although some are experimenting with such algorithms. Most cite that this comes down to a lack of explainability and transparency, an increase in overall technicality, a lack of transferability and the fact that they see more value in streamlining other, simpler technologies. Near-future prospects, however, seem to point in a much larger adoption of ML and possibly DL algorithms, as practical knowledge of said algorithms grows and explanation techniques develop.

Thirdly, we have, to some extent, validated existing theories and concepts about XAI in a practical environment. Most literature lacks insight into AI-application in the practical environment. It has been our specific goal to test theories and concepts against actual employees who have pragmatic needs and demands. This was done through the mentioned types of interviews that were conducted.

Lastly, in this thesis, we have attempted to unify the terminology and concepts pulled from existing literature to give a clear and complete breakdown of XAI. A handful of deep dives into the subject of XAI have been done in recent years. Although each of these papers has tackled the topic broadly and in-depth, several papers use different terminology to describe explainability and transparency. This mostly comes down to the fact that the field of XAI is relatively new and, thus, not enough time has been given to exchange findings among scholars.

7.3 Limitations and future work

As with any study, some things could be done to further improve the design research carried out. For example, while interviews were conducted to learn about the adoption and prospects of AI and XAI techniques in the Dutch insurance industry as a whole, more interviews would give a more complete view of the industry. This way we could be more confident that the sample chosen gives an accurate representation of the entire population. The same can be said for the confirmatory interviews. Although the addition of more confirmatory interviews would presumably not change any substantial elements of the general design, it would rather refine the smaller details a bit further.

Additionally, since industry experts employed at insurance companies and software suppliers were targeted for the interviews, end-customers, consumers and lawmakers were not consulted. This should not be a big problem, considering the checklist was not meant for these demographic groups specifically, but would nonetheless have possibly given some additional insight.

Moreover, only a small number of companies in the industry are now starting to gain traction with ML and DL concepts. This means that knowledge of the technologies among industry experts is still relatively scarce. Considering, that in the future this knowledge will grow, more detailed analyses could be done on the topic.

To combat this knowledge gap and stay relevant to all types of applications used in the insurance industry, our analysis was generally broad in including all relevant AI-techniques, ranging from Rule-Based Systems to Convoluted Neural Networks. Because of this, a more detailed approach diving into, for example, a single algorithm, would presumably give a more explicitly conclusive and unambiguous analysis for that specific algorithm.

Conversely, the aforementioned broadness comes with a benefit. Even though the checklist is meant for application in the financial sector, the checklist could potentially also be of similar use in other sectors, as we were careful not to restrict to any particular type of AI system. After all, there aren't many competing approaches that this checklist design can be compared with, although the 'Independent High-Level Expert Group on Artificial Intelligence', which was set up by the European Commission, has published a 'Trustworthy AI Assessment List'[2] in July 2020, which includes nine questions dedicated to explainability.

Lastly, as initially mentioned, the checklist is meant to extend an existing checklist, named 'Checklist-KOAT', which is made by SIVI. Specifically, the design mentioned in this thesis serves as a base for the eventual integration into the 'Checklist-KOAT'. This integration will be done by SIVI itself. SIVI will keep improving the integrated design through field testing with associated member companies. We presume the design will remain relevant for the foreseeable future, although, as time progresses and new techniques become prevalent, eventual updates will inevitably be advisable.

[2]The 'Trustworthy AI Assessment List' can be found at https://ec.europa.eu/digital-single-market/en/news/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment

References

Barocas, S. and Selbst, A. D. (2016). Big data's disparate impact. *California Law Review*, 104:671.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82 – 115.

Castelvecchi, D. (2016). Can we open the black box of AI? *Nature*, 538(7623):20–23.

Daley, S. (2019). 32 examples of AI in healthcare that will make you feel better about the future. https://builtin.com/artificial-intelligence/artificial-intelligence-healthcare. Retrieved on 08.02.2021.

European Union (2010). *Charter of Fundamental Rights of the European Union*, volume 53. European Union, Brussels.

Haynes, S. R., Cohen, M. A., and Ritter, F. E. (2009). Designs for explaining intelligent agents. *International Journal of Human-Computer Studies*, 67(1):90 – 110.

Hevner, A. R. (2007). The three cycle view of design science research. *Scandinavian Journal of Information Systems*, 19(2):87–92.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

Hofman, E. (2018). A brief history of machine learning models explainability. https://zelros.medium.com/a-brief-history-of-machine-learning-models-explainability-f1c3301be9dc. Retrieved on 13.02.2021.

Kroes, M. (2020). Fact-AI: A framework towards responsible AI. https://www.viqtordavis.com/en-us/media/whitepaper-fact-ai. Whitepaper from Viqtor Davies, Retrieved on 10.11.2020.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Preece, A., Harborne, D., Braines, D., Tomsett, R., and Chakraborty, S. (2018). Stakeholders in explainable ai.

Rakova, B., Yang, J., Cramer, H., and Chowdhury, R. (2020). Where responsible ai meets reality: Practitioner perspectives on enablers for shifting organizational practices. *ArXiv*, abs/2006.12358.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, volume abs/1602.04938, pages 1135–1144.

Ruf, B., Boutharouite, C., and Detyniecki, M. (2020). Getting fairness right: Towards a toolbox for practitioners.

Schreiber, D. (2019). AI can vanquish bias: Algorithms we can't understand can make insurance fairer. https://www.lemonade.com/blog/ai-can-vanquish-bias/. Retrieved on 01.03.2021.

Sheh, R. and Monteath, I. (2017). Introspectively assessing failures through explainable artificial intelligence. *Introspective Methods for Reliable Autonomy*.

Smuha, N. A. (2019). The EU approach to ethics guidelines for trustworthy artificial intelligence. *Computer Law Review International*, 20(4):97–106.

Tjoa, E. and Guan, C. (2020). A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, page 1–21.

Tomsett, R., Braines, D., Harborne, D., Preece, A., and Chakraborty, S. (2018). Interpretable to whom? a role-based model for analyzing interpretable machine learning systems. *arXiv preprint arXiv:1810.00184*.

United Nations (2015). Transforming our world: The 2030 agenda for sustainable development. A/RES/70/1. Resolution adopted by the General Assembly on 25 September 2015.

Vilone, G. and Longo, L. (2020). Explainable artificial intelligence: a systematic review.

Zednik, C. (2019). Solving the black box problem: A normative framework for explainable artificial intelligence. *Philosophy Technology*.

A The checklist

Quality checklist for AI applications in the insurance industry

Transparency & explainability

Transparency refers to how understandable the inner workings of the application are and how well its individual components and their interrelationships can be viewed. By **explainability**, we mean the way an application conveys to the user (from customer to expert) in a human-understandable way, how the input leads to the results (i.e. output), which in turn can lead to novel or confirming insights about the model and its dataset. Both aspects ensure that the application as a whole becomes more robust. It can help increase the accuracy of your model, justify its functionality, prevent unwanted biases, uncover new knowledge and help prevent and correct errors.

#	Subject	Check	Answer	Elucidation
1	Transparency of algorithm	Elaborate whether the application itself is already transparent to the user or whether external techniques are needed to increase transparency and explainability.	Open answer	We can divide AI algorithms into two categories. Namely, *transparent algorithms* and *non-transparent algorithms*. Transparent algorithms include *rule-based systems,linear/logistic regression, decision trees, k-nearest neighbours, rule-based learners, general additive models, bayesian models.* Non-transparent algorithms include *tree ensembles, support vector machines* and *neural networks.* It is also possible that your application falls somewhere between the two extremes.
2	Purpose	Which of these reasons is most important with regard to the application's explainability. You can choose more than one option.	Multiple choice	The need to explain a model mainly stems from one of four reasons: 1) increasing the model's accuracy. 2) discovering or confirming causality. 3) verifying and justifying the model's fairness & robustness. 4) Checking the model for errors and removing bugs.
3	Development	Explain how the right balance has been found between accuracy and explainability in the development process and where priority has been placed.	Open answer	AI applications are often developed with the highest possible accuracy as the main priority. In such a case, a compromise is usually made between accuracy on the one hand and transparency on the other. In practice, explainability and transparency are often just as important and sometimes even more important for the use of an application.
4	Impact	1) Explain what the results of the application are used for (i.e. what role do the results fulfil). Are the results advisory to the user or does the application make autonomous decisions based on these results, or something in between the to ends? 2) Explain how the application's result influences the end-customer.	Open answer	The application's impact on the organization is influenced by the role that the application's result must fulfil. If the application plays an advisory role, the result's impact on the end customer is relatively less than if the results play a decisive role (meaning the application makes autonomous decisions based on the results). If the application makes autonomous choices and implements incorrect logic, incorrect decisions can go unnoticed. After which they can only be corrected
5	Explainability by design	1) What external techniques have been used to improve explainability by design? 2) Explain how these techniques increase the explainability of the application.	Open answers	There are external techniques that can increase the transparency and explainability of AI applications. Some of these techniques involve baking in explainability from the beginning. This has to do, for example, with paying extra attention to input processing or training on the dataset. These techniques are called *ante-hoc techniques*. Think of techniques such as *Reversed Time Attention Model (RETAIN)*, *Bayesian deep learning* (*BDL*), etc.
6	Add-on explainability	1) What external techniques have been used to improve add-on explainability? 2) Explain how these techniques increase the application's explainability.	Open answer	There are external techniques that increase an application's explainability after or during the model run-time. These techniques are called *post-hoc techniques*. There are post-hoc techniques that are universal for all algorithm types, but also for specific algorithms types. Think of techniques such as *Local Interpretable Model-Agnostic Explanations (LIME), SHapley Additive exPlanations (SHAP), Layer-wise Relevance Propagation (LRP)*, etc.
7	Explanation output type	Explain what type of explanation is outputted?	Open answer	The outputted explanation of the application can be given in several types. The possible options are *textual, numeric, categorical, pictorial, time series or rule-based.*
8	Stakeholders	For each relevant stakeholder, explain... 1) what demands they have concerning the application's explainability and 2) how these interests are fulfilled by the application's explanations:	Open answer	Different people in and outside the organization have different demands concerning the application's explainability. The organisation must take these different demands into account, so that an application is as robust and usable as possible, while also considering ethical concerns and legislation. (possible stakeholders: *creator, examiner, operator, executor, decision-subject, data-subject*) (possible demands: *informativeness, causality, trustworthiness, confidence, accessibility, interactivity, transferability, privacy-*
9	Redress (Question and complaint handling)	Explain how the consumer receives more information when he or she has an in-depth question or complaint regarding the result.	Open answer	If the application affects consumers, they may sometimes need further explanation about their situation. In such a case, the application's results may have to be explained to this person. This can happen, for instance, if the application makes a mistake, or the consumer has a
10	Bias	Explain... 1) which end-customer groups may be unfairly disadvantaged by the application and how this is prevented. 2) whether the explanations provided reveal (unwanted) biases (in the data or the algorithm)?	Open answer	Undesired biases can arise in applications that are trained on data. These unwanted biases must be actively prevented. Explainability and transparency can be a means to that end. Example of bias: Statistically, red cars take more damage. However, does this mean that red car owners have to pay a higher premium?
11	Expertise	Explain whether new expertise concerning explainability and ethics were needed in the company, since the implementation of the application?	Open answer	Some companies hire 'explanation experts' and 'ethics experts' to support their AI projects.