SN Computer Science (2024) 5:81
https://doi.org/10.1007/s42979-023-02401-z

ORIGINAL RESEARCH

Deceptive XAI: Typology, Creation and Detection

Johannes Schneider[1] · Christian Meske[2] · Michalis Vlachos[3]

Received: 21 April 2023 / Accepted: 9 October 2023

Abstract
Providing rationales for decisions can enhance transparency and cultivate trust. Nevertheless, in light of economic incentives and other factors that may encourage manipulation, the reliability of such explanations comes into question. This manuscript builds upon a previous conference paper* by introducing a conceptual framework for deceptive explanations and constructing a typology grounded in interdisciplinary literature. The focus of our work is on how AI models can generate and detect deceptive explanations. In our empirical evaluation, we focus on text classification and introduce modifications to the explanations generated by GradCAM, a well-established method for explaining neural networks. Through a user study comprising 200 participants, we demonstrate that these deceptive explanations have the potential to mislead individuals. However, we also demonstrate that machine learning (ML) techniques can discern even subtle deceptive tactics with an accuracy exceeding 80%, given sufficient domain expertise. Furthermore, even in the absence of domain knowledge, unsupervised learning can be employed to identify inconsistencies in the explanations, provided that fundamental information about the underlying predictive model is accessible.

Keywords Explainability · Artificial intelligence · Deception · Detection · Typology

Introduction

Artificial intelligence (AI) holds the potential to enhance global prosperity and improve overall well-being. However, concerns regarding its various applications persist, with one significant worry being the lack of adequate oversight of online content, which has resulted in the proliferation of deceptive information. For example, online media platforms constantly grapple with the challenge of combating the spread of "fake news". Similarly, e-commerce platforms dedicate substantial resources to identify and remove misleading product reviews. Notably, there exist marketing strategies that deliberately generate fraudulent reviews, aiming either to exaggerate the virtues of products or to misrepresent their quality [62]. Generative AI models, such as large language models like ChatGPT, have the potential to exacerbate the issue of deception [18, 56].

Due to the limited moderation of online content, various attempts at deception are on the rise. Online media platforms are grappling with the pervasive issue of disseminating "fake news," while e-commerce websites are investing considerable efforts to detect and counter deceptive product reviews (see [62] for an extensive review). Some marketing strategies even involve the deliberate creation of fake reviews to artificially enhance product perceptions or make false claims about product quality [3].

There are multiple motives for providing "altered" explanations for the functioning of predictive AI systems. Providing entirely truthful explanations may risk exposing the underlying logic of the AI system, including its intellectual

This article is part of the topical collection "Advances on Agents and Artificial Intelligence" guest edited by Jaap van den Herik, Ana Paula Rocha and Luc Steels.

This is an extended journal version of a conference paper [53]. It adds a conceptualization of deceptive explanations and contains updates to related work.

✉ Johannes Schneider
johannes.schneider@uni.li

Christian Meske
christian.meske@ruhr-uni-bochum.de

Michalis Vlachos
michalis.vlachos@unil.ch

[1] University of Liechtenstein, Fuerst Franz Josef Strasse, 9490 Vaduz, Liechtenstein

[2] University of Bochum, Bochum, Germany

[3] HEC, University of Lausanne, Lausanne, Switzerland

Published online: 09 December 2023

property. Decision-makers may also choose to deviate from AI-generated recommendations at their discretion. For example, a bank employee might deny a loan to a disliked potential client, citing an AI model's recommendation as the basis, supported by a fabricated explanation, regardless of the system's actual recommendation.

AI systems might achieve improved performance by leveraging information that should not be used but is nonetheless available. For instance, private health information about an individual could be utilized by insurance companies to accept or reject applicants. Even though such practices are prohibited in certain countries, the information remains highly valuable in estimating the expected costs associated with applicants. Additionally, product recommendations delivered through recommender systems often come with explanations [17], intended to boost sales. Companies may be inclined to provide explanations that entice customers into making purchases, irrespective of their accuracy.

Consequently, there are incentives to develop systems that employ such information covertly, concealing their use of "illicit" decision criteria from authorities or even citizens. In Europe, the GDPR law grants individuals the right to access explanations for decisions made through automated processes. Such legal initiatives have been introduced with the intent of addressing the adverse aspects of AI. Nevertheless, given the evolving nature of AI and, more specifically, the field of explainable AI (XAI) [33], there remains a limited understanding of both explainability and the potential for deception through explanations.

This paper builds upon a previous conference version by incorporating a conceptual framework that delves into the realm of deceptive XAI (Deceptive XAI—Typology, Creation and Detection). In summary, this paper presents several notable contributions:

1. Conceptualization: The paper makes a valuable contribution by offering a comprehensive conceptualization of the problem, drawing upon prior research. This contribution encompasses the development of a typology for categorizing deceptive explanations and the introduction of an explainee model tailored for understanding deceptive explanations. Recognizing that deceptive AI involves both technical and human dimensions, the paper incorporates insights not only from computer science but also from related fields such as information systems and social sciences, particularly addressing the human aspects. The typology serves the additional purpose of identifying research gaps and providing guidance for future researchers.
2. Deception mechanisms evaluated with a user study: The paper focuses its attention on deceptions arising from the manipulation of relevance scores, a particularly relevant area given that explanations often rely on relevance scores. This aligns with common practices in various XAI techniques, including popular approaches like LIME, SHAP, and GradCAM. Through empirical analysis, including a user study, the paper reinforces prior findings by demonstrating that deceptive AI explanations can indeed mislead individuals.
3. Formal analysis of detection boundaries: The paper conducts a formal, theoretical analysis that establishes generic conditions for detecting deceptive explanations. It highlights the essential role of domain knowledge in uncovering certain forms of deception that may be beyond the grasp of explainees (i.e., recipients of explanations).
4. Detection algorithms: We contribute to the ongoing efforts to combat deceptive explanations by introducing both supervised and unsupervised detection methods. These methods represent an initial step in the pursuit of detecting deceptive explanations. Furthermore, the paper underscores that the success of deception detection hinges on various factors, including the specific type of deception, the availability of domain-specific knowledge, and a fundamental understanding of the deceptive system.

Conceptualization of Deceptive Explanations

In this section, we begin by establishing the foundational context for explanations and the concept of deception. We then proceed to formulate a comprehensive typology of deceptive explanations, drawing upon insights from a range of academic disciplines. Additionally, we introduce an explainee model that explores different stimuli and their corresponding responses in the context of deceptive explanations. It is crucial to acknowledge that explanations are ultimately intended for human consumption, underscoring the importance of comprehending the mechanisms governing how humans perceive and process AI predictions and their associated explanations.

Explanation and Deception

Several definitions of explanation and deception exist, but the ones particularly relevant to the perspective of deceptive XAI are as follows:

- Explanation:"An explanation is the details or reasons that someone gives to make something clear or easy to understand." [13].
- Deception: "A communicator's deliberate attempt to foster in others a belief or understanding which the communicator considers to be untrue." [12, p. 1553].

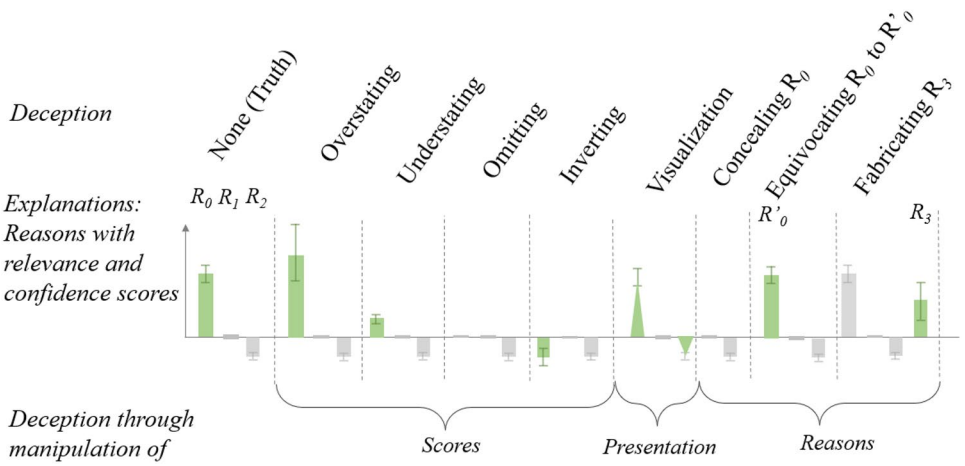

Fig. 1 Deviations of explanation from the truth. Each reason is accompanied by a relevance and confidence score

Two crucial facets emerge from these definitions. First, deception is inherently an intentional action. This implies that actions resulting in user misguidance due to inadvertent technological flaws or lack of knowledge are not categorized as deceptive. Second, deception manifests through the transmission of information, which is essentially a form of communication. In the context of deceptive explanations, this pertains to systems utilized by individuals to obtain information, encompassing information systems and human-computer interaction, given that explanations are fundamentally directed at a human audience.

The individual in the role of the *explainer*, responsible for providing explanations, typically possesses knowledge that the *explainee*, who is the recipient of the explanation, does not have. Consequently, there exists an inherent information asymmetry between these two parties. Specifically, the explainer possesses a certain level of insight into the accuracy and truthfulness of the explanations. Within our context, *an explanation is comprised of a collection of rationales used to elucidate specific outcomes of an AI (referred to as local explanations) or the AI as a whole (referred to as global explanations)*. A *reason* is a piece of information providing clarification through contextual, causal, or other knowledge. It might be expressed as simple facts or complex decision rules. Reasons are potentially accompanied by two pieces of auxiliary information provided by the explainer, namely: a relevance estimate of the reason and the certainty of the estimate. The relevance estimate serves to quantify the degree of impact attributed to each rationale.

In the context of AI, a comprehensive understanding of explanations hinges on the elucidation of the *information* that forms the foundation of the explanation process. This information delineates the scope within which alterations and fabrications may occur. In the context of AI-driven explanations, the explanation method draws upon three primary sources of information: the model's inputs, the model's outputs, and the model itself. Inputs can be further distinguished into data that altered the model, i.e., training data used to fit model parameters and test data possibly used during deployment. Outputs of the model typically constitute a decision, but for generative models, they might also consist of an artifact such as a photorealistic image, a piece of music, etc. The model consists of a model definition, called architecture in deep learning, and fitted parameters. Other, less prevalent auxiliary information used by an XAI method is information on the explainee, e.g., to personalize explanations, and contextual information, e.g., on how training data is obtained, or meta-data, e.g., a description of attributes in the training data.

Typology

To derive a typology of deceptive explanations, we investigate both technical and non-technical aspects. We assess the state of XAI based on recent surveys [1, 33, 54], i.e., how explanations are typically computed and presented. In addition, we build on generic theory describing how explanations emerge as deceptive from a human perspective, e.g., [64] describe deceptive information practices in human-computer interaction focusing on an E-commerce setting. The combination of both viewpoints helps in a more comprehensive treatment. The final typology shown in Fig. 1 highlights three deception mechanisms. The first, altering scores, originates from existing XAI techniques. Existing literature on XAI treats explanations commonly as a set of fixed reasons accompanied by *relevance scores*. Relevance scores quantitatively capture how important a reason is compared to others, which allows for comparing reasons. For example, for attribution-based techniques, GradCAM, LIME, and SHAP compute relevance scores of input attributes but do so in a different manner. Reasons are mostly fixed and simple, e.g., reasons could be the presence or absence of input attributes. For example-based XAI methods that state the most influential samples from the training data, a reason

Table 1 Overview of deception types based on reasons

Deception Type	Description	Example: Explanation of a Recommendation
Equivocation	Reasons are vague or ambiguous, but not clearly incorrect	Deception: Some of your friends were also interested in the product recently. Truth: Two friends looked for information on resolving issues with the product, but none showed interest in purchasing
Concealment	Reasons are withheld, omitted, or disguised	Deception: Most people liked the product. Truth: While this might be true, the system might conceal that most of your friends explicitly disliked it
Falsification	Reasons might be fabricated or altered. Fabricated reasons are made-up for the sole purpose of deception. It might be impossible for a model to decide based on fabricated reasons, since, for example, it lacks the information stated in the fabricated reasons as inputs. Alteration modifies reasons so that they potentially still bear similarity with the original reason but can be clearly identified as being untrue	Deception: Some of your friends were also interested in the product. (i) The system has no information on who your friends are (Fabrication). (ii) The system possesses information on friends. But only brief acquaintances were interested in the product

could also be the presence of a sample in the training data. A natural extension to reporting relevance scores is adding a *confidence score* to each relevance score to capture uncertainty in the estimate. An explainability method typically defines a *measure*, commonly in the form of an algorithm that determines how relevance scores are computed. Deception can alter scores. That is, a truthful explanation might be altered by changing relevance and confidence scores. A trivial mechanism simply sets scores to zero, which corresponds to hiding that a specific reason played a role. Scores might also be altered indirectly by constructing or modifying XAI algorithms.

The second dimension is motivated from a non-technical perspective of the deception of humans as discussed in [64]. The three general types of deception from [64] shown in Table 1 are concealment, falsification, and equivocation. They focus on deception due to new or altered reasons rather than scores. That is, while score-based deception treats reasons as fixed, reason-based deception mechanisms might fabricate reasons that could potentially impossibly be relevant. For example, the argument that a bird was classified as a specific breed based on her singing, cannot be true for a classifier that is only trained on images. A reason-based explanation might also conceal reasons, e.g., omit them as shown in Fig. 1. However, there is a conceptual difference compared to score-based omission. In score-based omission, the system might output a score of zero, which implies that the reason seems relevant though not for the specific instance. In reason-based omission, if a reason is not mentioned it appears to be as irrelevant as all other non-mentioned reasons.

The third dimension relates to presentation, e.g., explanations might be visualized [36] or stated in a way that is misleading.

Explainee Model of Deceptive Explanations

In Fig. 2 we provide a theoretical model for deceptive explanations from the perspective of the explainee, i.e. the person receiving the prediction and explanation. Since there is little conceptual work on deceptive explanations in AI, it is mainly a synthesis of works on deception and explanation, particularly, it builds on models by [64] and [19]. [64] provides the overarching framework in the context of deception in information systems, i.e., systems used by people to acquire information. [19] provides the perceptions triggered by explanations. More precisely, [64] describes the instantiation of the generic stimulus-organism-response framework to the context of deceptive information on e-commerce websites. We adopt this framework but alter sections as required for deceptive XAI. The "approach behavior toward target AI outcome", i.e., toward predictions and explanations, is defined as the positive attitude or action toward them. An approach behavior toward target is observed when the explainee shows an attitude towards the (deceptive) prediction and explanation that she would not have generated for a truthful explanation. This attitude, in turn, might lead to positive actions judged by the deceiver. Thus, a deception attempt is successful, if a person shows such a behavior. For instance, if a person buys a product as a reaction to a deceptive recommendation accompanied by a deceptive explanation.

The individual, problem, and situational characteristics refer to factors that impact the chances of success of a deception attempt. Individual factors cover traits of a person such as being critical or adherence to authority (e.g., there is considerable interpersonal variance in credibility assigned to automatically generated information [61]) and personal relevance (e.g., does the prediction and explanation impact a person strongly?). Individual factors also cover domain knowledge of an explainee, i.e. knowledge

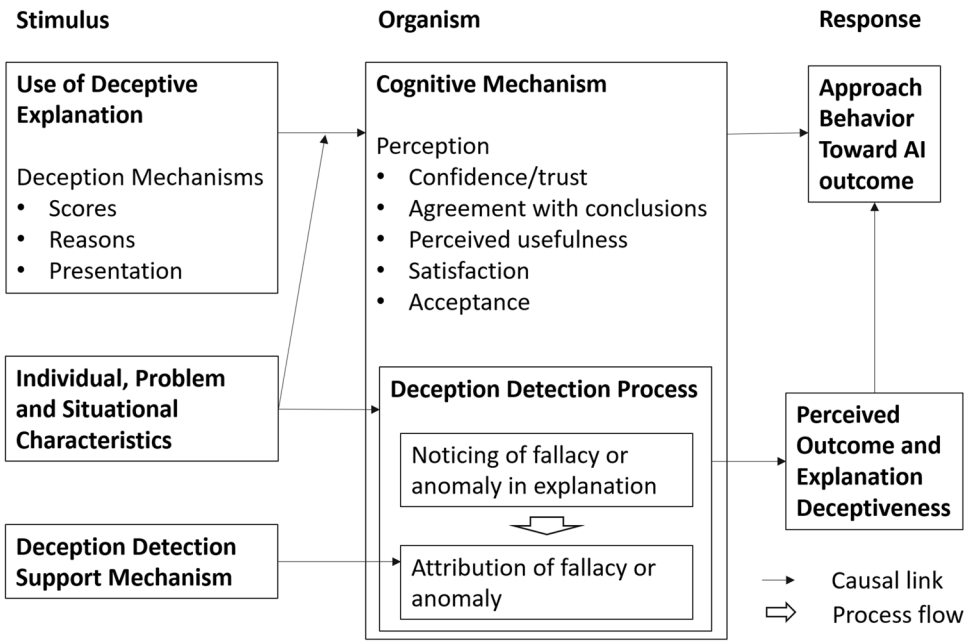

Fig. 2 Theoretical Model of Deceptive Explanations

of data the AI operates on, and technical expertise in AI technology. For instance, a doctor is more likely to discover poor explanations of a medical diagnosis than an ordinary person. Problem characteristics refer to the problem the AI addresses. Complexity of the problem, available information on the problem, etc. play a vital role in the ability to detect deception. Situational factors involve available resources to investigate the explanation. For instance, an auditor or a government body might have more resources in terms of accessible competencies as well as finances than a private person. Deception detection support mechanisms refer to any mechanisms that assist in detecting deceptions. Such mechanisms might include regulations with respect to system design and governance. For example, reporting and transparency standards prescribing the structure of explanations and their required level of detail can facilitate detection.

The cognitive mechanisms describe the cognitive processes through which a deceptive explanation impacts an explainee. Cognitive mechanisms consist of beliefs, thoughts, or perceptions about predictions and explanations, the AI system, or the instance that can be held responsible for the AI system or putting it in use. [19] put forth the following perceptions of intelligent systems, which are (still) valid: confidence/trust in judgments, agreement with conclusions, perceived usefulness, satisfaction, and acceptance. Deceptive explanations might result in deviations from an explainee's preconceived expectation as well as observed violations of sound reasoning. These instill a two-step deception detection process: identifying anomalies in explanations, which often involves investigating the explanation on its own as well as with respect to the prediction and the input to the AI. Anomalies primarily aim at the verification of factual claims or processes for which an explainee has preconceived expectations. In addition, explanations to support deceptions might suffer from fallacies that occurred during the reasoning process, such as "the use of invalid or otherwise faulty reasoning" [58]. Humans are capable of numerous techniques to attack fallacies [11]. As a consequence of the detection process, an explainee might believe that the explanation, as well as the prediction, are manipulated and the AI system as a whole is deceptive. "Use of deceptive explanation" refers to the employment of one or more types of deceptive manipulations of a (truthful) explanation, based on the typology presented in Fig. 1.

Problem Definition: Deception Through Explanations and Predictions

Explanations should help in understanding a model. This understanding is based on providing reasons that lead to (reported) predictions of a model. A reported prediction is the output class shown to the user. Explanations have to be judged together with the model prediction. For global explanations of a classification model, these might be general rules or concepts guiding the decision-making process of the model. For instance, a general rule could be that an object is classified as a car if four tires are detected. For local explanations, these are the specific concepts or rules for a given input that caused the output of the model. Thus, there are two sources of deception for an explainee: model decisions and explanations. Explanations can either be truthful to the model predictions or not. If an explanation is truthful for a reported output it answers the question:

"What are the reasons that have caused or would have caused the model to make the reported prediction?"

More formally, an (AI) model M maps input $X \in S$ to an output Y, where S is the set of all possible inputs. We introduce a *reference model* M^* and a *reference explainability method* H^*. In practice, M^* might be a deep learning model and H^* a commonly used explainability method such as GradCAM, LIME, or SHAP. That is, H^* might not be perfect. Still, we assume that the explainee trusts it, i.e. she understands its behavior and in what ways explanations differ from "human" reasoning. The model M^* is optimized with a benign objective, for example, maximizing accuracy. We assume that M^* is not optimized to be deceptive. However, model M^* might not be fair and behave unethically. The explainee could be a layman with limited AI knowledge who assumes that decisions are made following M^*, and explanations are based on H^* though she is generally not aware of model behavior and lacks access to training data. She might also not understand model decisions and only possess limited knowledge of the explainability method H^*. A deceiver might pursue objectives other than those used for M^* leading to the deceiver's model M^D. The model M^D might simply alter a few decisions of M^* using simple rules or it might be a completely different model. A (truthful) explainability method $H(X, Y, M)$ receives input X, class label Y, and model M to output an explanation. For the reference explanation method H^*, this conforms to providing a best-effort, ideally, a truthful, reasoning, why model M would output class Y. The deceiver's method H^D might deviate from H^* using arbitrary information. It returns $H^D(X)$, where the exact deception procedure is defined in context. In particular, $H^D(X)$ might simply modify the reference explanation. For example, it might first internally compute $H^*(X)$ and omit certain reasons. The method $H^D(X)$ might also compute an adversarial sample X' based on X and return the explanations $H^*(X')$. The adversarial sample might trick the reference explainability method into outputting deceptive explanations. An explainee (the recipient of an explanation) obtains for an input X, a decision $M^D(X)$, and an explanation $H^D(X)$. The decision is allegedly from M^* and the explanation allegedly from H^* and truthful to the model M^D providing the decision. Thus, an explainee should be lured into believing that $M^*(X) = M^D(X)$ and $H^D(X) = H^*(X, M^D(X), M^D)$. However, the deceiver's model might or might not output $M^D(X) = M^*(X)$. A deceiver might also choose an explainability method H^D that differs from H^*, or she might explain a different class Y.

The *goal of a deceiver* is to construct an explanation so that the explainee is neither suspicious about the decision in case it is truthful nor to the model M^D, that is, $M^D(X) \neq M^*(X)$, nor to the explanation $H^D(X)$ if it deviates from $H^*(X, M^D(X), M^*)$. Thus, an explanation might be used to hide an unfaithful decision from the model or it might be used to convey a different decision-making process than occurs in M^D.

Deception Scenarios

Deception of an explainee can involve either model decisions or explanations or both. This leads to multiple scenarios that we describe next. An explanation $H^*(X, Y, M)$ provides truthful reasons why model M would yield output Y for an input X. We shall first consider explanations for the model M^D and the input X, where outputs Y have only one of two values, i.e., we use $H^*(Y) := H^*(X, Y, M^D)$ with $Y \in \{M^D(X), M^*(X)\}$. This yields four scenarios shown in Fig. 4. These scenarios judge the truthfulness of explanations only based on whether they are aligned with the reported prediction, i.e., the output class shown to the user. That is, a prediction is also considered truthful if it explains why the model M^D would output $M^*(X)$ differing from the actual model prediction $M^*(X) = M^D(X)$. However, this view is somewhat simplistic. An explanation for the actual prediction, i.e., $H(M^*(X))$, yields valuable information on how the input X is processed by the model M^D, i.e., what are the reasons and relevance scores extracted when the model processes input X. This holds even if the reported prediction differs, $M^*(X) \neq M^D(X)$. The explanation $H^*(X, M*(X), M^D)$ might be considered truthful to the model and the input (though not for the reported prediction). Furthermore, consider an explanation $H^*(X', Y', M^D)$ for an input X with $X' \neq X$ (differing from the input X to explain) and an output Y' differing also from both the reported class $M^D(X)$ and the class $M^*(X)$ given by the reference model. Even in this case, one might argue that the explanation is more truthful than a random prediction. For example, for structured input data such an explanation still (truthfully) reveals what attributes are relevant for the reference model M^* for some input X'.

In a more general case, one might assume that an arbitrary model M^D, differing from the reference model M^* is used, that an arbitrary explainability method H^D is used, or that the reference method H^* yielding truthful explanations is used. If H^* is used inputs might be altered, i.e., explanations being truthful to reported predictions Y^R, model predictions for the input $M^D(X)$, input X and the model itself M^D. For example, there might be a reference model M^* and a deceiver model M^D such that M^* was trained using training data without gender and race as attributes and model M^D included them. For explanations, model M^* is used, while reported (more accurate) predictions stem from M^D. This leads to 32 combinations covering different aspects of truthfulness. We only show an extension of the four basic scenarios in Fig. 3

Truthful?				Description
Reported pred. $\mathbf{M^D(X^D)}$ = pred. of reference model $\mathbf{M^*(X^D)}$?	Explanation $\mathbf{H^*(\cdot,\cdot)}$ for			
	reported prediction $\mathbf{M^D(X)}$? $\mathbf{H^*(\cdot,M^D(X))}$?	input $\mathbf{X}$? $\mathbf{H^*(X,\cdot)}$?	Truthful explanation for	
Y	Y	Y	$\mathbf{H^*(X,M^D(X))}$	Telling the truth
Y	Y	N	$\mathbf{H^*(X',M^D(X'))}$	(Reported) prediction is truthful. Explanation is unfaithful since $X' \neq X$
Y	N	Y	$\mathbf{H^*(X, Y')}$	Prediction is truthful. Explanation is unfaithful since prediction $Y' \neq M^D(X)$, but it is truthful with respect to input X
Y	N	N	$\mathbf{H^*(X', M^D(X'))}$	Prediction is truthful. Explanation is unfaithful.
N	Y	Y	$\mathbf{H^*(X,M^D(X))}$	Prediction is unfaithful. Explanation is truthful.
N	Y	N	$\mathbf{H^*(X', M^D(X))}$	Prediction is unfaithful. Explanation is unfaithful since input $X' \neq M^D(X)$, but it is truthful with respect to prediction
N	N	Y	$\mathbf{H^*(X,Y')}$ $\mathbf{Y' \neq M^D(X)}$	Prediction is unfaithful. Explanation is unfaithful since prediction $Y' \neq M^D(X)$
N	N	N	$\mathbf{H^*(X',Y')}$ $\mathbf{Y' \neq M^D(X)}$	Both prediction and explanation are unfaithful.

Fig. 3 Scenarios for reported predictions and explanations taking also truthfulness to inputs into account

by taking truthfulness with respect to inputs into account (see Fig. 3). The motivation is that inputs are commonly subject to manipulation to deceive classifiers (and humans), i.e., manipulated inputs are known as adversarial samples. That is, a deceiver might generate samples X', e.g., by altering samples X through adversarial manipulations, though that reference explainability method H^* reports deceptive explanations. She might report the output for model $M^D(X)$, but use another sample X' to compute the explanation. We say that input is truthful if the same input is used to compute $M^D(X)$ and to compute the explanation. If an explanation is truthful to the input, i.e., based on input X, it answers the question:

"What are the reasons (including their scores) by the model to process the input and make its prediction (though not necessarily the reported one)?"

Score-Based Deceptive Explanations

In this section we conceptualize and analyze score-based deception, i.e., we provide a problem definition, measures to capture the degree of deception and discuss the creation and detection of deceptive explanations by manipulating relevance scores. This constitutes one of the four manipulation mechanisms (see Fig. 1). We focus on deception through relevance scores because we deem it the most feasible

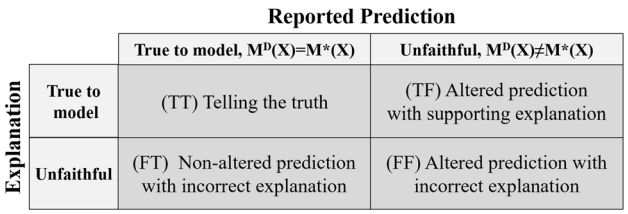

Fig. 4 Scenarios for reported predictions and explanations

approach to be conducted in an automated manner. The usage of scores operationalizes measures for deception, i.e., to express them in intuitive mathematical terms that require only limited explanations and assumptions.

Problem Definition

We consider classification systems that are trained using a labeled dataset $\mathcal{D} = \{(X, Y)\}$ with two sources of deception: model decisions and explanations. Definition of the reference and deceiver explainability methods H^*, H^D and models M^*, M^D are provided in "Problem definition: deception through explanations and predictions". Here, we focus specifically on the four scenarios (see Fig. 4). We write $H^*(X) := H^*(X, M^D(X), M^D)$. An input X consists of values

for n features, $\mathcal{F} = \{i \mid i = 1 \dots n\}$, where each feature i has a single value $x_i \in V_i$ of a set of feasible values V_i. For example, an input X can be a text document such as a job application, where each feature i is a word specified by a word id x_i. Documents $X \in S$ are extended or cut to a fixed length n. ML models learn (a hierarchy of) features. Explaining in terms of learned features is challenging since they are not easily mapped to unique concepts that are humanly understandable. Thus, we focus on explanations that assign *relevance scores* to features $\mathcal{F}$ of an input X.

Formally, we consider explanations H that output a value $H_i(X, Y, M)$ for each feature $i \in F$. Where $H_i > 0$ implies that feature i with value x_i is supportive of decision Y. A value of zero implies no dependence of i on the decision Y. $H_i < 0$ shows that feature i is indicative of another decision.

Measuring Explanation Faithfulness

We measure the faithfulness of an explanation using two metrics, namely *decision fidelity* and *explanation fidelity*.

Decision Fidelity

Decision fidelity amounts to the standard notion of quantifying whether input X and explanation $H^D(X)$ on their own allow deriving the correct decision $Y = M^*(X)$ [48]. Therefore, if explanations indicate multiple outputs or outputs different from Y, this is hardly possible. Decision fidelity f_D can be defined as the loss when predicting the outcome $Y = M^*(X)$ of a model M^* measured using another classifier g based on the reported explanations $H^D(X)$ and inputs X only, or formally:

$$\begin{aligned} &f_D(X) = -L(g(X, H^D(X)), Y) \\ &\text{with classifier } g : (X, H^D(X)) \mapsto \{0, 1\} \end{aligned} \tag{1}$$

The loss might be defined as 0 if $g(X, H^D(X)) = Y$ and 1 otherwise. We assume that the reference explanation $H^*(X, M^*(X), M^*)$ results in minimum loss, i.e., maximum decision fidelity. (Large) decision fidelity does not require that an explanation contains all relevant features used to derive the decision $M^D(X)$. For example, in a hiring process, gender might influence the decision, but for a particular candidate other factors, such as qualification, social skills, etc., are dominant and on their own unquestionably lead to a hiring decision.

Explanation Fidelity

Explanation fidelity refers to the overlap of the (potentially deceptive) explanation $H^D(X)$ and the reference explanation $H^*(X, M^D(X), M^D)$ for an input X and reported decision $M^D(X)$. Any mismatch of a feature in the two explanations lowers explanation fidelity. It is defined as:

$$f_O(X) = 1 - \frac{\|H^*(X, M^D(X), M^D) - H^D(X)\|}{\|H^*(X, M^D(X), M^D)\|} \tag{2}$$

Even if the decision $M^D(X)$ is non-truthful to the model, i.e., $M^D(X) \neq M^*(X)$, explanation fidelity might be large if the explanation correctly outputs the reasoning that would lead to the reported decision. If the reported decision is truthful, i.e., $M^D(X) = M^*(X)$, there seems to be an obvious correlation between decision- and explanation fidelity. However, any arbitrarily small deviation of explanation fidelity from the maximum of 1 does not necessarily ensure large decision fidelity and vice versa. For example, assume that an explanation from H^D systematically under- or overstates the relevance of features, i.e. $H^D(X)_i = H^*(X)_i \cdot c_i$ with arbitrary $c_i > 0$ and $c_i \neq 1$. For c_i differing significantly from 1, this leads to explanations that are far from the truth, which is captured by low explanation fidelity. However, decision fidelity might yield the opposite picture, such as maximum decision fidelity, since a classifier g (Definition 1) trained on inputs $(X, H^D(X))$ with labels $M^D(X)$ might learn the coefficients c_i and predict labels without errors.

Explanation fidelity captures the degree of deceptiveness of explanations from H^D by aggregating the differences of its relevances of features and those of the reference explanations. When looking at individual features from a layperson's perspective, deception can arise due to over- and understating the feature's relevance or even fabricating features (see Fig. 5). Omission and inverting of features can be viewed as special cases of over- and understating. In this work, we do not consider feature fabrication.

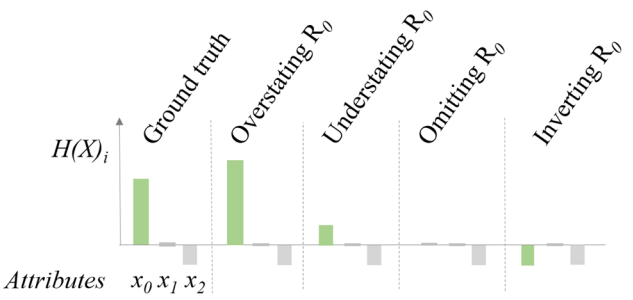

Fig. 5 Deviations of relevance scores from (trusted) reference explanation

Creation of Deceptive Explanations

We first discuss goals a deceiver might pursue using deceptive explanations, followed by how deceptive explanations can be created using these goals in mind.

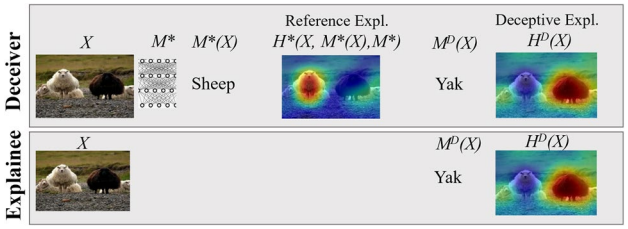

Fig. 6 Inputs and outputs for deceiver and explainee for scenario FT in Fig. 4. Images by [39]

Purposes of Deceptive Explanation

(i) Convincing the explainee of an incorrect prediction, i.e. that a model decided Y for input X although the model's output is $M^D(X)$ with $Y \neq M^D(X)$. For example, a model M^* in health-care might predict the best treatment for a patient trained on historical data $\mathcal{D}$. A doctor might change the prediction. She might provide the best treatment for well-paying (privately insured) patients and choose a treatment that minimizes her effort and costs for other patients.

(ii) Providing an explanation that does not accurately capture model behavior without creating suspicion. An incorrect explanation will manifest in low decision fidelity and explanation fidelity. It involves hiding or overstating the importance of features in the decision process (Fig. 5) with more holistic goals such as: a) Omission: Hiding that decisions are made based on specific attributes such as gender or race to prevent legal consequences or a loss of reputation. b) Obfuscation: Hiding the decision mechanism of the algorithm to protect intellectual property.

The combination of (i) and (ii) leads to the four scenarios shown in Fig. 4. The most intricate scenario is providing a prediction differing from the model using a non-truthful explanation (FF). For example, a college official might opt to admit a student to a program because of a bribe although the model would not favor her admission. The model might see as a strong reason for rejecting poor grades and as a reason for accepting a good financial situation of the applicant. In scenario FF, the explanation might, for instance, overstate features that were not relevant, claiming that the student's age is a reason for acceptance. If the explanation was aligned with the (wrong) decision (TF), the explanation might list reasons the model provides for accepting her, i.e., how the model would explain the acceptance decision. In the example, this could be a good financial situation.

Creation

To construct deceptive explanations (and decisions), a deceiver has access to the model M^* and M^D, the input X, and the reference explanation H^*. She outputs a decision $M^D(X)$ in combination with an explanation $H^D(X)$ (see Fig. 6). Deceptive explanations are constructed to maximize the explainee's credence of decisions and explanations. We assume that an explainee is most confident that the reference explanation $H^*(X, Y, M^D)$ and the model-based decision $Y = M^*(X)$ are correct. This encodes the assumption that the truth is most intuitive since any deception must contain some reason that can be identified as faulty.

We provide simple means for creating deceptive explanations that are non-truthful explanations (FT and FF). The idea is to alter reference explanations. This approach is significantly simpler than creating deceptive explanations from scratch using complex algorithms as done in other works [3, 4, 26], while at the same time guaranteeing high-quality deceptive explanations since they are based on what the explainee expects as a valid explanation. For non-truthful explanations a deceiver aims at over-, understating, or omitting features $X' \subseteq X$ that are problem- or instance-specific. To obtain non-truthful explanations we alter reference explanations in two ways:

Definition 1 (Omission) Remove a fixed set of values $\mathcal{V}$ so that no feature i has a value $x_i \in \mathcal{V}$ as follows:

$$H_{Omit}(X)_i := \begin{cases} 0, & \text{if } x_i \in \mathcal{V}. \\ H^*(X)_i, & \text{otherwise.} \end{cases} \tag{3}$$

In our context, this means denying the relevance of some words $\mathcal{V}$ related to concepts such as gender or race. The next alteration distorts relevance scores of all features, for example, to prevent re-engineering through obfuscation.

Definition 2 (Noise addition) Add noise in a multiplicative manner for any explanation $H^*(X)$:

$$H_{Noise}(X)_i := H^*(X)_i \cdot (1 + r_{i,X}), \tag{4}$$

where $r_{i,X}$ is chosen uniformly at random in $[-k, k]$ for a parameter k for each feature i and input $X \in S$.

We assume that these alterations are applied consistently for all outputs. Note, that this does not imply that all explanations are indeed non-truthful, e.g., for noise it might be that by chance explanations are not altered or only very little. For omission it might be that a feature is not relevant in the decision for a particular input X, i.e., the value of a feature $H^*(X)_i$ is zero in this case.

Deception Detection

To detect deception attempts, we reason using explanations and decisions of multiple inputs. That is, for a set of inputs $X \in S^D$, we are given for each input X the reported decision $M^D(X)$ and accompanying explanation $H^D(X)$. Our goal is to identify whether a model outputs deceptive explanations. For supervised learning, we (even) aim to identify the inputs yielding deceptive outputs. We assume that only features that are claimed to contribute positively to a decision are included in explanations. Features that are claimed to be irrelevant or even supportive of another decision outcome are ignored. The motivation is that we aim at explanations that are as simple to understand as possible. The omission of negatively contributing features makes detection harder. We first provide theoretical insights before looking into practical detection approaches.

Formal Investigation

Ideally, any of the three types of deception $\{TF, FT, FF\}$ is detected using only one or more inputs $X \in S^D$ and their responses $M^D(X)$ and $H^D(X)$ (see Fig. 6). But, without additional domain knowledge (such as correctly labeled samples), metadata, or context information, this is impossible for all deception attempts. This follows since data, such as class labels, bear no meaning on their own. Thus, any form of "consistent" lying is successful, e.g. always claiming that a cat is a dog (using explanations for class dog) and a dog is a cat (using explanations for class cat) is non-detectable for anybody lacking knowledge of cats and dogs, i.e., knowing what a cat or a dog is.

Theorem 1 *There exist non-truthful reported decisions $M^D(X) \neq M^*(X)$ that cannot be identified as non-truthful.*

Proof Consider a model M^D for dataset $\{(X, Y)\}$ for binary classification with labels $Y \in \{0, 1\}$ and $M^D(X) = M^*(X)$. Assume a deceiver switches the decision of model M^*, i.e. it returns $M^D(X) = 1 - M^*(X)$ and $H^*(X, M^D(X), M^D)$. Consider a dataset with switched labels, i.e. $\{(X, 1 - Y)\}$ and a second model M'^D that is identical to M^D except that it outputs $M'^D(X) = 1 - Y = 1 - M^D(X)$. Thus, reference explanations are identical, i.e. we have $H^*(X, M'^D(X), M'^D) = H^*(X, M^D(X), M^D)$. Thus, for input X both the deceiver model M^D and model M'^D report $M'^D(X) = M^D(X) = 1 - M^*(X)$ and $H^*(X, M^D(X), M^D)$. Therefore, M'^D and M^D cannot be distinguished by any detector. That is, if a detector reported M^D as deceiving it would also do so for M'^D, which is not deceiving but trained on a dataset with switched labels. □

Similarly, non-truthful and truthful explanations $H \neq H^*$ might be indistinguishable, when using decision fidelity as a measure for truthfulness, e.g. using feature inversion $H(X) = -H^*(X)$. Consider two explainability methods H and H^*, where one is the inverse of the other, i.e. $H(X)_i = -H^*(X, Y)_i$. Both allow to achieve the same decision fidelity, e.g. a classifier $g : (H(X, Y), X) \rightarrow Y$ and $g' : (H^*(X, Y), X) \rightarrow Y$ can obviously achieve the same accuracy. Thus, without any domain knowledge, both explanation methods H and H^* appear equally truthful with respect to decision fidelity.

The following theorem states that one cannot hide that a feature (value) is influential if the exchange of the value with another value leads to a change in decision.

Theorem 2 *Omission of at least one feature value $v \in \mathcal{V}$ can be detected, if there are instances $X, X' \in S$ with decisions $M^D(X) \neq M^D(X')$ and $X' = X$ except for one feature j with $x_j, x'_j \in \mathcal{V}$ and $x'_j \neq x_j$.*

Proof We provide a constructive argument. We can obtain for each input $X \in S$, the prediction $M^D(X)$ and explanation $H^D(X)$. By Definition of omission, if feature values $\mathcal{V}$ are omitted it must hold $H^D(X)_i = 0$ for all $(i, X) \in \mathcal{F}_{S,v}$ and $v \in \mathcal{V}$. Omission occurred if this is violated or there are $X, X' \in S$ that differ only in the value $x_j \in \mathcal{V}$ for feature j and $M^D(X) \neq M^D(X')$. The latter holds because the change in decision must be attributed to the fact that of $x_j \neq x'_j$, since X and X' are identical except for feature j with values that are deemed omitted. □

Theorem 2 is constructive, meaning that it can easily be translated into an algorithm by checking all inputs S if the stated condition is matched. But, generally all inputs S cannot be evaluated due to computational costs. Furthermore, the existence of inputs $X, X' \in S$ that only differ in a specific feature is not guaranteed. However, from a practical perspective, it becomes apparent that data collection helps in detection, i.e. one is more likely to identify "contradictory" samples X, X' in a subset $S' \subset S$ the larger S' is.

Detection Approaches

Our formal analysis showed that only decisions and explanations are not sufficient to detect deception involving flipped classes. That is, some knowledge on the domain is needed. Encoding domain know-how with a labeled dataset seems preferable to using expert rules or the like. Thus, not surprisingly, this approach is common in the literature, e.g. for fake news detection [38, 41]. To train a detector, each sample is a triple $(X, M^D(X), H^D(X))$ for $X \in S^T$ together with label

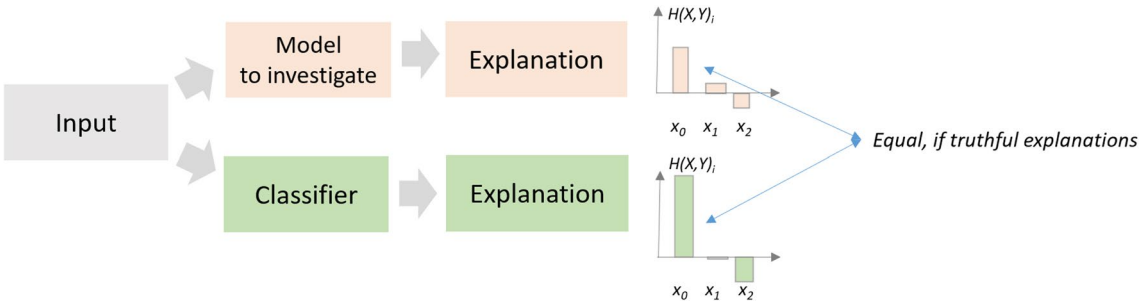

Fig. 7 Illustration of Algorithm *ConsistencyChecker*. It is based on comparing explanations of the model to investigate and another classifier, serving as a reference model. If explanations of both models match well (for all samples), explanations are not deceptive

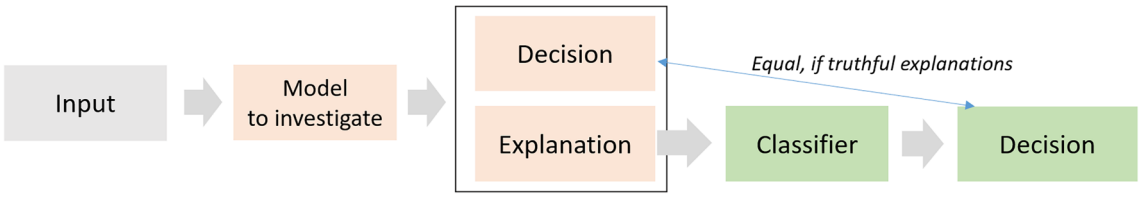

Fig. 8 Illustration of Algorithm *UsefulnessChecker*. It is based on comparing decisions of the model to investigate and a classifier (reference model) that obtains explanations as inputs. If decisions are equal (for all samples), explanations are not deceptive

Algorithm 1 Consistency Checker

Input: Untrained models $\mathcal{M}'$, reference method H^*, inputs S^D with (deceptive) decisions and explanations $\{(M^D(X), H^D(X)\}$
Output: (Outlier) Probability p
$S^{M'} = s$ randomly chosen elements from S^D with s random in $[c_0|S^D|, |S^D|]$
(We used: $c_0 = 0.33$*)*
Train each model $M' \in \mathcal{M}'$ on $(X, M^D(X))$ for $X \in S^{M'}$
$m_i(X) = \frac{1}{|\mathcal{M}'|} \sum_{M' \in \mathcal{M}'} H_i^*(X, M^D(X), M')$
$s(M') = \frac{\sum_{i \in [0,n-1], X \in S^D} (H_i^*(X, M^D(X), M') - m_i(X))^2}{n|S^D|}$
$s(M^D) = \frac{\sum_{i \in [0,n-1], X \in S^D} (H_i^D(X) - m_i(X))^2}{n|S^D|}$
$\mu = \frac{1}{|\mathcal{M}'|} \sum_{M' \in \mathcal{M}'} s(M')$
$\sigma = \frac{1}{|\mathcal{M}'|} \sqrt{\sum_{M' \in \mathcal{M}'} (s(M') - \mu)^2}$
$p = prob\big(T > |s(M^D) - \mu| \;\big|\; T \sim \mathcal{N}(0, \sigma)\big)$

$L \in \{TT, FT, TF, FF\}$ stating the scenario in Fig. 4. After training, the classifier can be applied to the explanations and decisions of $X \in S^D$ to investigate. We develop classifiers maximizing deception detection accuracy.

Labeling data might be difficult since it requires not only domain knowledge of the application but also knowledge of ML, i.e. the reference model and explainability method. Thus, we also propose unsupervised approaches to identify whether a model, i.e. its explanations, are truthful to the model decision. That is, the goal is to assess if given explanations H^D are true to the model $M^D(X)$ or not.

Our first approach is to check, whether the explanations of H^D and decisions of M^D are consistent (see Fig. 7). This would be easy, if the model M^D was available, i.e. we would check if $H^*(X, M^D(X), M^D) = H^D(X)$. Since it is not, we aim to use a model M' to approximate model M^D and compare the explanations H^* of M' with H^D. Since approximation introduces an error, we must determine if differences in

Algorithm 2 Usefulness Checker

Input: Untrained models $\mathcal{M}'$, reference method H^*, inputs S^D with (deceptive) decisions and expl. $\{(M^D(X), H^D(X)\}$, untrained classifier model C
Output: (Outlier) Probability p
$S^{M'} = s$ randomly chosen elements from S^D with s random in $[c_0|S^D|, |S^D|]$
We used: $c_0 := 0.33$
Train each model $M' \in \mathcal{M}'$ on $(X, M^D(X)$ for $X \in S^{M'}$
S^T = random subset of S^D of size $c_1|S^D|$ *We used:* $c_1 := 0.8$
$C^{M'}$ = trained classifier model C on $\big(H^*(X, M^D(X), M'), M^D(X)\big)$ for $X \in S^T$ and $M' \in \mathcal{M}'$
C^{M^D} = trained classifier model C on $\big(H^D(X, M^D(X), M^D), M^D(X)\big)$ for $X \in S^T$
$Acc(C^M) :=$ Accuracy of classifier C^M using $X \in S^D \setminus S^T$
$\mu = \frac{1}{|\mathcal{M}'|} \sum_{M' \in \mathcal{M}'} Acc(C^{M'})$
$\sigma = \frac{1}{|\mathcal{M}'|} \sqrt{\sum_{M' \in \mathcal{M}'} (Acc(C^{M'}) - \mu)^2}$
$p = prob\big(T > |Acc(C^{M^D}) - \mu| \mid T \sim \mathcal{N}(0, \sigma)\big)$

the explanations originate from model approximation or from deception. To do so, we train (approximate) reference models $M' \approx M^D$ with $M' \in \mathcal{M}'$ using the provided data $(X, M^D(X))$ with $X \in S^D$. The models might differ, e.g. in hyperparameter settings. If explanations of M^D are "outliers" relative to those of models $M' \in \mathcal{M}'$ explanations might be untruthful and further investigation is needed. Otherwise, any variation of explanation compared to H^* due to deception is comparable to variation that might arise if models are slightly varied but no deception is attempted.

Details on the model M^D are unknown. For many common ML tasks, excellent performing architectures are publicly known. Therefore, we assume that we can anticipate at least a well-working architecture for the problem – the closer it is to M^D the better.

Algorithm 1 (*ConsistencyChecker*) shows details. It takes as input a set of untrained models $M' \in \mathcal{M}'$ – ideally similar to M^D. The more information is known on M^D, the more similar the models can be chosen. If only very little information or no information on M^D is available, one might train models on $(X, M^D(X))$ for $X \in S^D$ and choose those that perform best.

The second unsupervised approach Algorithm 2 (*UsefulnessChecker*) tries to reproduce the decisions based on explanations (see Fig. 8). If this is not possible for explanations from M^D but for those of approximate models M^D then explanations from M^D are likely deceptive. As before, we train multiple approximate models $M' \in \mathcal{M}'$. For each model M', we train a classifier $C^{M'}$ using explanations from the approximate models $\mathcal{M}\simeq$ as well as one on explanations from M^D. We use the same classifier architecture for all. We conduct a statistical test (as in Algorithm 1), if accuracy is an outlier. The full pseudo-code is similar to Algorithm 1. For the sake of completeness, it is shown in Algorithm 2.

Evaluation

We elaborate on two text classification tasks using a convolutional neural network (CNN) for text classification by [22] as our reference model M^* and GradCAM [55] for generating reference explanations H^*. The CNN is well-established, conceptually simple, and works reasonably well. GradCAM was one of the methods said to have passed elementary sanity checks that many other methods did not [2]. Note that neither our creation nor deception approaches are dependent on the actual XAI method, i.e., we only require and use the relevance scores produced by (any) XAI method but do not depend on the method internals. While GradCAM is most commonly employed for CNN on image recognition the mechanisms for texts are identical. In fact, [27] showed that GradCAM on CNNs similar to the one by [22] leads to outcomes on human tasks that are comparable to other explanation methods such as LIME. The GradCAM method, which serves as a reference explanation H^*, computes a gradient-weighted activation map starting from a given layer or neuron within that layer back to the input X. We apply the reference explanation method H^*, i.e., GradCAM, on the neuron before the softmax layer that represents the class Y' to explain. For generating a high fidelity explanation for an incorrectly reported prediction $M^D(X) \neq M^*(X)$ (scenario FT in Fig. 4) we provide as explanation the reference explanation, i.e. $H^D(X) = H^*(X, M^D(X), M^D)$. By definition reference explanations maximize explanation fidelity f_O.

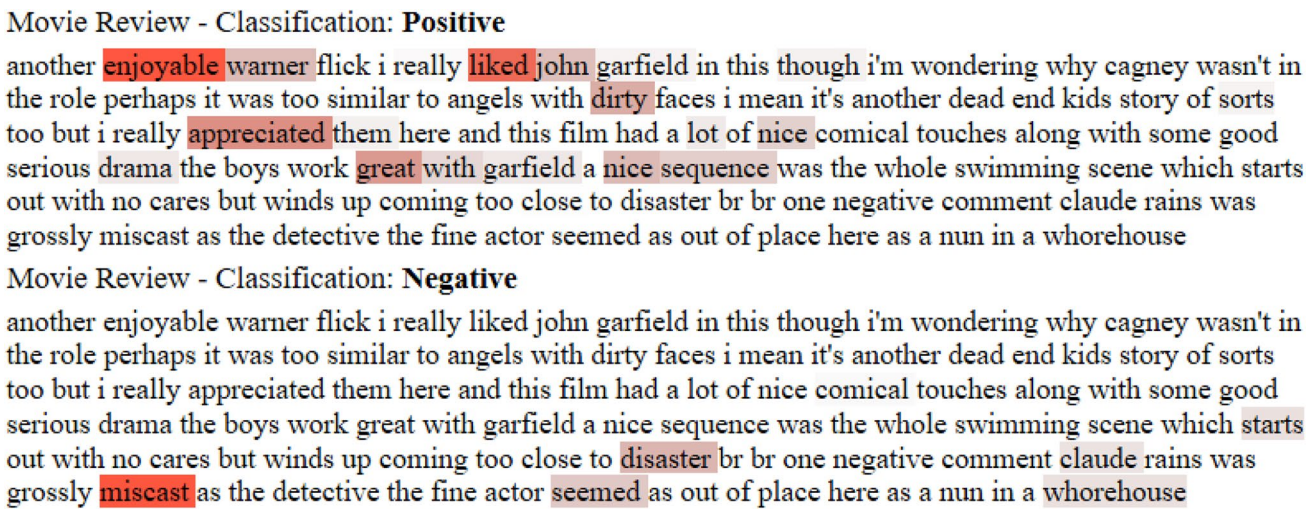

Fig. 9 Generated sample explanations for scenarios TT (top) and FT (bottom) from Fig. 4

Setup and Datasets

We employed two datasets. The IMDB dataset [30] consists of 50,000 movie reviews and a label indicating positive or negative sentiment polarity. We also utilized the Web of Science (WoS) dataset consisting of about 47,000 abstracts of scientific papers classified into 7 categories [23]. Our CNNs for classification achieved accuracies of 87% for IMDB and 75% for WoS trained with 2/3 of the samples for training and 1/3 for testing. We computed explanations for test data only. For deception using omission, we removed a randomly chosen set of words V (see Definition 1), such that their overall contribution to all explanations H^* is $k\%$ (with a tolerance of $0.01k$ %). The contribution of a word v is given by $\sum_{(i,X)\in\mathcal{F}(v,S)} H^*(X, M^*(X))_i$. For explanation distortion parameter k (see Definitions 1 and 2) we state values for each experiment.

ML-Based Detection

As detector models, we used CNN models. For supervised learning, the model input is a concatenation of three vectors: (i) a text vector of word indices, (ii) a heatmap vector of values obtained via GradCAM, which is a 1:1 mapping of the visual output shown to the user, and (iii) a one-hot prediction vector of the decision. Our "simple" CNN detector, i.e. classifier, is designed as follows: we perform an embedding and concatenate the heatmap vector with the word embedding before doing a 1D convolution. Then we concatenate the one-hot prediction vector and use two dense layers. The more "complex" CNN adds six more conv1D layers: two processing the embedding, two on the heatmap vector, and two after the first concatenation. We used dropout for regularization. Since labeling is difficult and potentially error-prone, we consider different levels of label noise, i.e., $L \in [0, 0.32]$, such that a fraction L of all labels was replaced with a random label (different from the correct one). For the detection experiment, we chose samples that were predicted correctly by the truthful model. For unsupervised learning, we train 35 classifiers $M' \in \mathcal{M}'$ being variations of a CNN network [22], i.e., each of the following hyperparameters was chosen uniformly at random for each classifier M': embedding dimension $\{32, 64, 128\}$; 1–3 linear layers; 2–6 conv layers for the Kim network with a varying number of filters. We also varied the training sets in terms of size and elements, i.e. we trained a model with a subset of $\mathcal{T}$ of size 33, 50, and 100%. All models were trained using the Adam optimizer for 100 epochs. Train/Test data split was 80/20 for all detector models.

Classifiers learning from (deceptive) explanations as done in our unsupervised approach *UsefulnessChecker* tend sometimes to focus on raw inputs X and disregard explanation relevance scores $H_i^D(X)$. That is, they often work well and show little variation in accuracy despite large variations in explanations. To avoid this, we convolve also an inner representation of the network with explanation values enforcing stronger entanglement. That is, in the *UsefulnessChecker* model the output of the word embedding of the input is convolved with the explanations as follows: First, we perform a low-dimensional embedding (just one dimensional) and multiply the embedding values with the explanation values and add explanation values on top. This is then fed into 3 Conv1D layers followed by two dense layers.

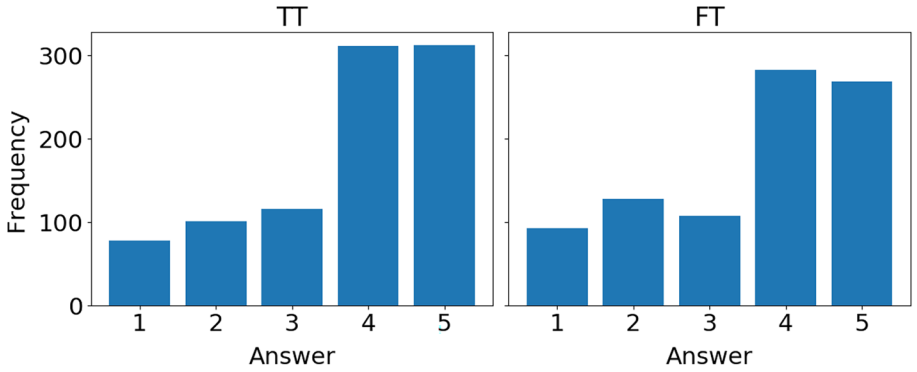

Fig. 10 Distributions of user replies to "The classification is correct" (1 = strongly disagree to 5 = strongly agree)

Table 2 Participants Demographics with $n = 140$ participants

Variable	Value	Percentage
Gender	Male	66%
	Female	34%
Age	≤ 25 years of age	18%
	From 26 to 40 years of age	62%
	From 41 to 65 years of age	18%
	> 65 years of age	2%
Education	High School	16%
	Associate Degree	11%
	Bachelor's Degree	56%
	Master's Degree	16%
	Doctoral Degree	1%

Human-Based Detection

We conducted a user study using the IMDB dataset.[1] For the scenarios of interest, we compare explanations that are aligned to the shown prediction, i.e. TT and FT. Two samples are shown in Fig. 9. We recruited a total of 200 participants on Amazon Mechanical Turk from the US having at least a high-school degree. We presented each participant with 25 predictions together with explanations. They had to answer "Is the classification correct?" on a scale of five ranging from strongly disagree to strongly agree. We randomized the choice of presented samples, i.e. we randomly chose a sample of the dataset and we randomly chose between scenarios TT and FT in Fig. 4.

Results

Human-Based Detection

Out of the 200 participants, we removed participants that spend less than 5 s per question, since we deemed this time too short to provide a reasonable answer. We also filtered out participants who always gave the same answer for all 25 questions. This left 140 participants amounting to 3500 answers. Demographics and answer distributions are in Fig. 10 and Table 2.

A t-test of means confirmed that the distributions differ significantly (p-value of 0.008), though the mean scores for "agreeing" of 3.74(TT) and 3.58(FT) show that in absolute terms differences are minor. This implies that while the majority of humans might be fooled oftentimes, they have the capability to collectively detect deceptive explanations.

Supervised ML-Based Detection

Figure 11 shows results for IMDB. Those for WoS are similar, shown in Fig. 12. The plot shows that for very minor deception (small k) detection is more difficult. Accuracy rarely tops 60%, which is fairly low given that 50% is the baseline for guessing. Accuracy is also more volatile given label noise. Beyond some threshold under all scenarios, accuracy exceeds 90% for correctly deciding whether a decision and/or explanation are deceptive. Accuracy worsens with greater label noise, though it is often better than guessing even for a noise level of 32%. Mostly the complex CNN (with more layers) outperforms the simple one with one exception: the scenario where the explanation is always truthful to the model, but the decision is not. Here, a simple function allows distinguishing deceptive (FT) from the non-deceptive scenario (TT). We observed that for the scenario with noise on non-truthful explanations for correct decisions (TFNoise) small noise levels can lead to very low test accuracies. Training accuracies were larger, but mostly close to random guessing, indicating that the network fails to learn proper conditions what values constitute a noisy relevance score and which do not.

The more extreme the distortion of explanations, i.e. the larger k, the higher detection accuracy. But even for minor modifications, i.e. adding just 1% of noise on the explanation and omitting only words contributing to 1% of the total explanation, accuracy exceeds 80% under low label noise ($L = 0.01$). The only case in which this is non-obvious is

[1] The WoS dataset seems less suited since it uses expert terminology that is often not held by the general public from which participants originate as found in [27].

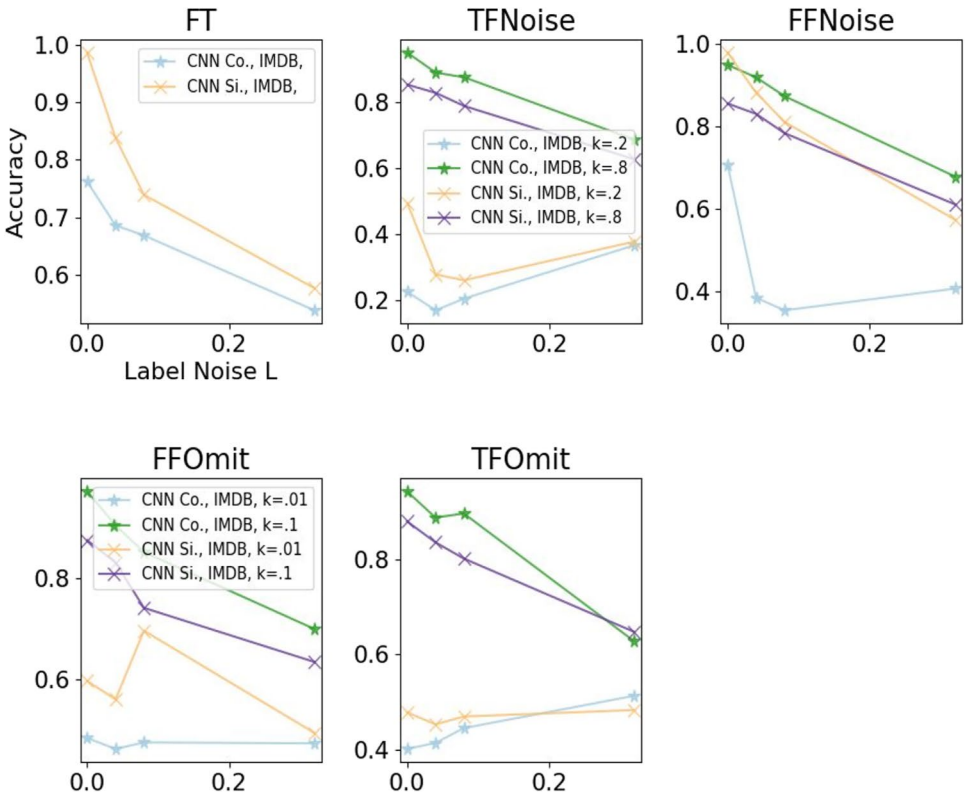

Fig. 11 Supervised detection results for IMDB for scenarios in Fig. 4

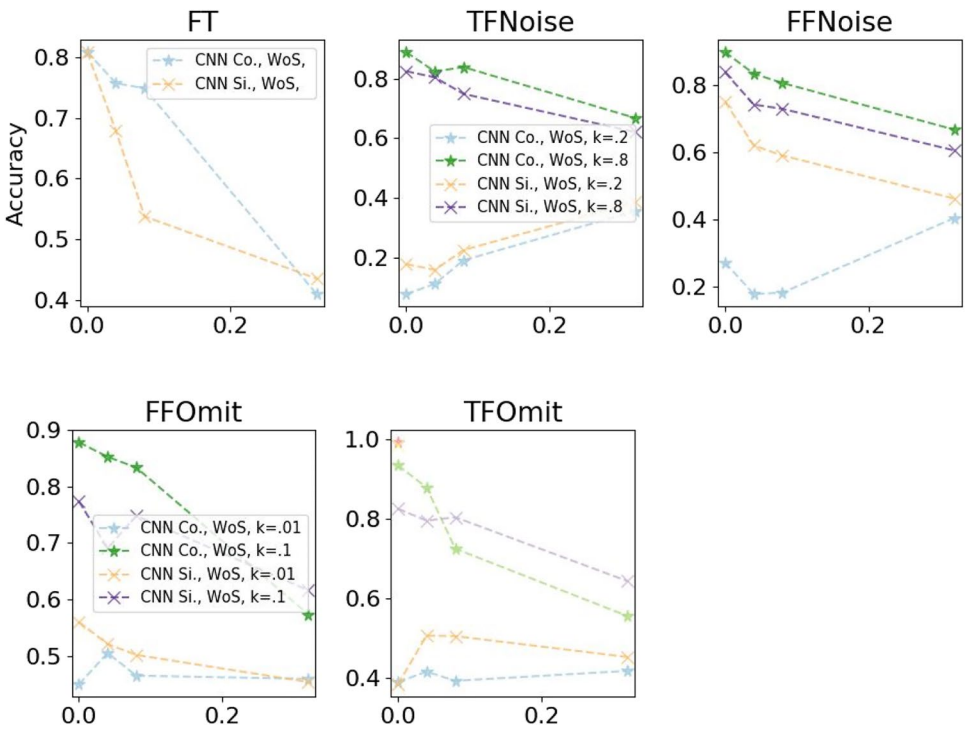

Fig. 12 ML-based supervised detection results for WoS for scenarios in Fig. 4

the scenario TFNoise (in Fig. 11), i.e. decisions are correct and explanations are not due to noise. In that case, for explanations altered with little noise, both classifiers are considerably worse than the trivial baseline of 50% accuracy. Both overfit the data in these cases. That is, training accuracies (not shown) are much higher than test accuracies.

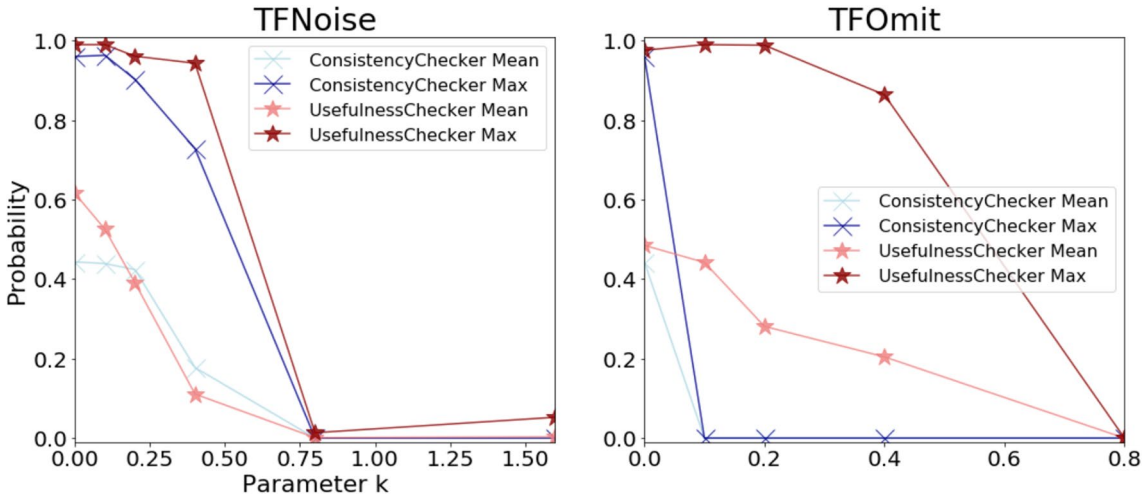

Fig. 13 Unsupervised detection results for IMDB

For larger noise levels ($k = 0.25$), random forests perform much better. CNNs also do well for $k = 1$, where test accuracies exceeded 90%. Comparing random forests and CNNs, CNNs seem to perform better in cases, where the prediction is incorrect. It seems that CNNs are able to learn the relation of inputs X to decision $M^D(X)$ without much need for an explanation. This is supported by the observation that the degree of deception manifested in the explanations has a limited impact on the performance. In contrast, random forests seem to rely more on explanations. They perform better in situations where decisions are truthful, but explanations are not.

Unsupervised ML-Based Detection

For a single model, a small probability indicates that it is unlikely to witness such an (extreme) outcome, i.e. accuracy or deviation from the average explanation. Thus, a small probability such as 0.05 indicates a high likelihood of deception. Figure 13 shows the average as well as the maximum probability of all 35 models. Detection ability follows an S-curve, meaning that for deception with parameterized strength k above a threshold detection works fairly well, whereas it fails otherwise. Required noise levels are somewhat higher for the WoS dataset, as shown in Fig. 12 but qualitatively, results are identical. The trained classifier performs worse on WoS than IMDB, since WoS is a more complex dataset. Classifiers vary (relatively) more in accuracy, which in turn also leads to more variance in explanations. Looking at the mean and max indicates that for multiplicative noise (TFNoise) all deceptive models are reliably detected for $k = 0.8$, while below that value at least for the most difficult-to-detect model it fails to do so. For $k < 0.8$, the detection methods are still valuable to identify suspicious models, meaning that such models exhibit lower probability, but are not low enough to be certain. The same reasoning also applies to TFOmit, though here a strong difference between methods is apparent. The *ConsistencyChecker* yields much better outcomes, highlighting that even small omissions can be detected reliably. It shows that statistical analysis is preferable to using a downstream task. Our models $\mathcal{M}_{\simeq}$ are very diverse, i.e. models differ by a factor of 3 in terms of training data and number of layers, as well as in neurons/filters per layer. We found that reducing (or increasing) the diversity has a profound impact on results, as shown in Figs. 14 and 15.

Difficulty of Deception Detection

We provide intuition for Algorithm *ConsistencyChecker* discussing the difficulty of detection depending on noise models and deception strategy. To compute the probability, we rely on values $s(M)$ as defined in Algorithm 1. We are interested in the gap $G(H_i, m_i(X)) := E[(H_i - m_i(X))^2]$ between the mean and the relevance score in the explanation of a feature i. For multiplicative noise we have $H_i^D(X, M^D(X), M) = (1 + U) \cdot H_i^*(X, M^D(X), M^D)$, where U is uniformly chosen at random from $[-k, k]$. We use $a_i := H_i^*(X, M^D(X), M^D)$ and $m_i := m_i(X)$ for ease of notation. We expect that the deviation for a deceptive explanation and the mean is:

$$\begin{aligned} G((1+U)a_i, m_i) &= E[((1+U)a_i - m_i)^2] \\ &= E[(a_i - m_i)^2 - 2Ua_im_i - U^2a_i^2] \\ &= (a_i - m_i)^2 + a_i^2k^2/3 \end{aligned}$$

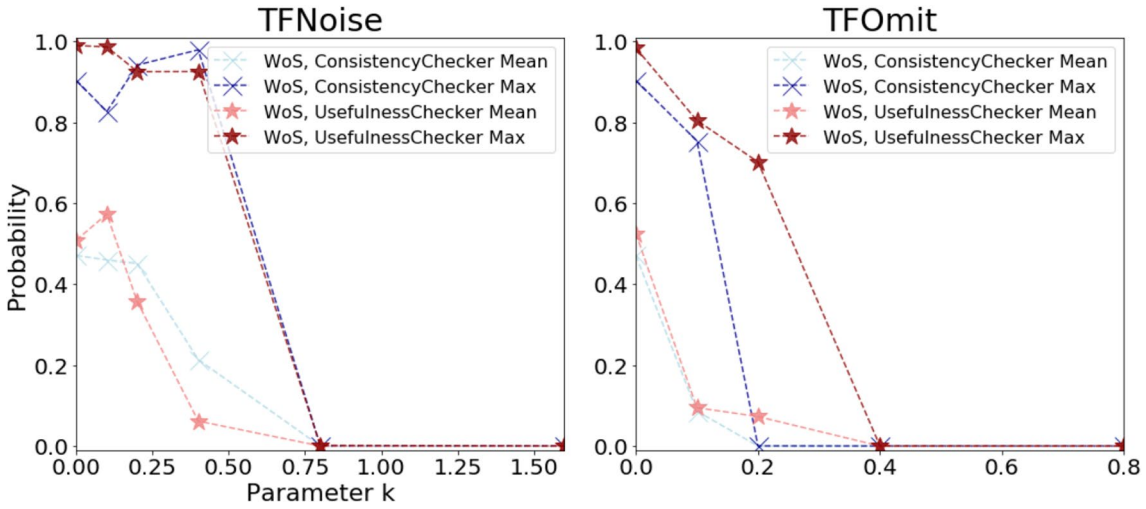

Fig. 14 Unsupervised detection results for WoS where approximate models vary only in training data (but have the same hyperparameters)

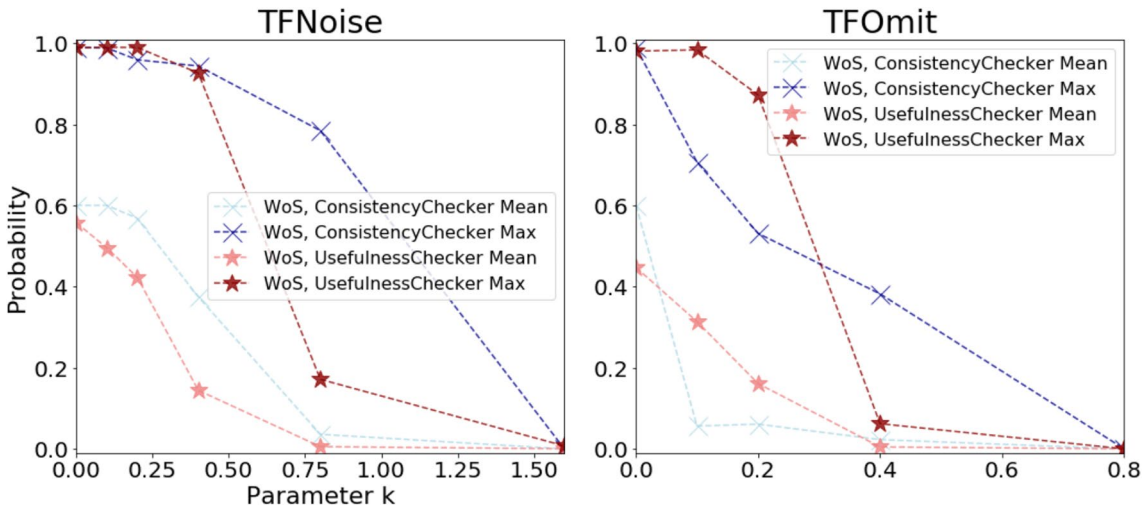

Fig. 15 Unsupervised detection results for WoS where approximate models vary in training data and hyperparameters. Detection is more difficult compared to varying training data only (Fig. 14)

The overall deviation $s(M)$ for a model is just the mean across all features i and inputs X. Detection is difficult when

$$\sum_{i,X}(a_i - m_i)^2 \gg \sum_{i,X} a^2k^2/3 \tag{5}$$

Put in words, detection is difficult, when the distortion due to deception (right-hand side term in Eq. 5) is small compared to the one due to model variations $\mathcal{M}\simeq$ (left-hand side term in Eq. 5). The closer a_i and m_i are and the larger k, the easier detection. If feature i is omitted then $G(a_i, m_i) = m_i^2$ and $G(a_i, m_i) = (a_i - m_i)^2$. Assume a set F^D of features is omitted, where the size of F^D depends on the parameter k. Deception is difficult if $\sum_i (a_i - m_i)^2 \gg \sum_{i\in F^D} m_i^2 + \sum_{i\notin F^D}(a_i - m_i)^2$. Clearly, the larger F^D the easier detection. Say we omit features with $m_i = a_i$ and we are given the choice of omitting two features with mean m or one with mean $2m$. The latter is easier to detect since means are squared, i.e. $m^2 + m^2 = 2\,m^2 < (2\,m)^2 = 4\,m^2$. Therefore, it is easier to detect few highly relevant omitted features than many irrelevant ones.

Related Work

Most commonly, deception aims at fooling machine learning models themselves. For example, adversarial examples [10] typically alter a data sample so that humans cannot recognize any differences, while the classifier is misled to output a class possibly chosen by the attacker. Few other works have also aimed at fooling both humans and classifiers, e.g., [45, 46] alter images in subtle ways so that both a classifier is confused and also a person assessing the modified sample. There have also been some works that aim at altering explanations to confuse persons and, possibly, also other machine learning models that might learn from explanations [51].

[21, 57, 60] showed how arbitrary explanations for methods relying on perturbations can be generated for instances by training a classifier with adversarial inputs. [14] trains a classifier using an explainability loss term for a feature that should be masked in explanations. [16] showed that biases in decision-making are difficult to detect in an input–output dataset of a biased model if the inputs were sampled in a way to disguise the detector. [25] used ML (including explanations) to support the detection of deceptive content. The explanations were non-deceptive.

[59] are interested in manipulating the inner workings of a deep learning network to output arbitrary explanations. Whether the explanations themselves are convincing, is not considered, i.e., the paper shows many examples of "incredible" explanations that can easily be detected as non-genuine. [4] focus on manipulating reported fairness based on a regularized rule list enumeration algorithm. [26] and [7] investigated the effectiveness of misleading explanations to manipulate users' trust. [26] used a model which decisions were made using prohibited features such as gender and race but misleading explanations were supposed to disguise their usage. Both studies [4, 7, 26] found that users can be manipulated into trusting high fidelity but misleading explanations for correct predictions. In contrast, we do not generate fake reviews but only generate misleading justifications for review classifications and provide detection methods and formal analysis. These initial, mostly empirical and algorithmic works provide interesting insights primarily on the creation of deceptive explanations of very specific problems and techniques. A formal analysis covering (at least) a wide class of problems and explainability methods has been missing. Furthermore, automatic deception detection has mainly been ignored as well as quantitative measures for deception. This work addresses these concerns.

Inspiration for detecting deceptive explanations might be drawn from methods used for evaluating the quality of explanations [24, 34]. In our setup, quality is a relative notion compared to an existing explainability method and not to a (human) gold standard. We compare the quality of reported explanations H^D to those of a reference explanation method H^* and not to a gold standard oriented towards humans. Our detection algorithms also relate to the field of AI forensics [47], since among other things it also aims to understand if an AI model was trained to misbehave. [37] investigated the influence of classifier accuracy and explanation fidelity on user trust. They found that accuracy is more relevant for trust than explanation quality though both matter. For three classifiers (differing strongly in test accuracy), they considered "random" explanations, i.e. using randomly chosen features, and "reference" explanations, i.e. explanations made by a (trustworthy) automatic method.

[35] investigated the impact of explanations on trust. Poor explanations reduce a user's perceived accuracy of the model, independent of its actual accuracy. Explanatory helpfulness varies depending on task and method [27]. Explanations are more helpful in assessing a model's predictions compared to its behavior. Some methods support some tasks better than others. For instance, LIME provides the most class discriminating evidence, while the layer-wise relevance propagation (LRP) method [6] helps assess uncertain predictions.

[3] showed how to create and detect fake online reviews of a pre-specified sentiment. In contrast, we do not generate fake reviews but only generate misleading justifications for review classifications. Fake news detection has also been studied [38, 41] based on ML methods and linguistic features obtained through dictionaries. [38, 41] use a labeled data set. Linguistic cues [29] such as flattery were used to detect deception in e-mail communication. We do not encode explicit, domain-specific detection features such as flattery. ML techniques have been used to detect lies uttered by humans in human interaction, e.g., [5].

Our methods might be valuable for the detection of fairness and bias – see [32] for a recent overview. There are attempts to prevent ML techniques from making decisions based on certain attributes in the data, such as gender or race [42] or to detect learnt biases based on representations [65] or perturbation analysis for social associations [40]. In our case, direct access to the decision-making system is not possible – neither during training nor during operations, but we utilize explanations.

In the context of human-to-human interaction using mainly audio-visual messages that 47% of lies are disclosed as deceptive and 61% of truths as non-deceptive [9]. [20] showed in a meta-review that even training only modestly improves detection rates. This makes the use of automatically generated explanations even more likely to succeed. Furthermore, cognitive human biases play a role and could be exploited to deceive humans through explanations [8].

In human-to-human interaction, behavioral cues such as response times [28] or non-verbal leakage due to facial expressions [15] might have some, but arguably limited

impact [31] on deception detection. In our context, this might pertain, e.g., to computation time. We do not use such information.

Discussion

Explanations introduce novel avenues for deception, as illustrated in Fig. 4. This trend is expected to intensify due to the increasing prevalence, creativity [52], and personalization [49] of AI technologies.

Deceptive explanations might aim at disguising the actual decision process, e.g., in case it is non-ethical, or make an altered prediction appear more credible. While faithfulness of explanations can be clearly articulated mathematically using our proposed decision and explanation fidelity measures, determining when an explanation is deceptive, is not always as clear, since it includes a grey area. An explanation might be deemed deceptive, though it might alternatively be labeled as merely inaccurate or simplified. Consequently, the task of detecting deception is a formidable challenge. Blatant forms of deception are readily identifiable, whereas more subtle variants are arduous to discern. Moreover, some degree of domain or model-specific knowledge proves indispensable. This could be data similar or, preferably, identical to the model's training data under investigation. Alternatively, domain experts could contribute insights in the form of labeled samples or detection guidelines, effectively evaluating model outputs and adjudicating their faithfulness or deception.

The task of identifying deceptive explanations becomes significantly more tractable when access to the model itself and its training or testing data is available. In such instances, the process essentially boils down to comparing model outputs against those suggested by the (training) data. We advocate for the enactment of regulatory measures that mandate auditors to possess genuine model access, thereby streamlining the process of deception detection. This represents a critical step towards ensuring that AI serves the broader societal good.

Unsupervised deception detection does not require any labeling of explanations as deceptive or non-deceptive. But it can require significant computation. Our algorithm 'ConsistencyChecker' requires a model similar to the model to investigate. That is both decisions and explanations must be similar. If no such (trained) model is available, it must be trained, which can be computationally expensive for large models found in generative AI, e.g., large language models. However, as deceivers are also likely to use or adapt (publicly available) pretrained models for the very same reason, this issue might not be such a concern in practice. Algorithm UsefulnessChecker only requires a model that is capable of predicting decisions from explanations well, which might be a simpler problem. That is, it is easier to identify a class, if one is aware of all relevant input features encoded in the explanation. In fact, it has been shown that learning with explanations can help improve classifier performance [51], but leveraging explanations in learning is still an active field of research.

Detection methods will improve, but so will strategies for lying. Thus, it is important to anticipate the weaknesses of detection algorithms that deceitful parties might exploit and mitigate them early on, for example, with the aid of generic security methods [43].The field of explainability is rapidly evolving, with a multitude of challenges on the horizon [33]. This dynamic landscape offers abundant prospects for future research, encompassing investigations into techniques for both crafting and identifying deceptive explanations. These endeavors may explore innovative avenues such as explanations pertaining to the features or layers of image processing systems, diverging from the traditional text-based explanations [50], or extend to the realm of multi-modal models [63]. Furthermore, the exploration of alternative models or architectures, such as foundation models [44], holds promise in enhancing our understanding and management of deceptive explanations in the AI landscape.

We limited our study to score-based deception mechanisms, which we deem more feasible than changing or fabricating reasons. However, given advances in the field of AI such deception mechanisms might become more practical.

Conclusion

In the realm of AI, a dynamic interplay between "liars" and "detectors" is emerging, driven by economic incentives and other motivating factors. Our work represents an initial move within this evolving game. We have structured the problem at hand and made a meaningful contribution by highlighting the inherent challenges associated with detecting deception attempts in the absence of domain-specific knowledge. Our machine learning models, enriched with domain expertise garnered from training data, exhibit good accuracy in detecting deception. We also showed that, unsupervised techniques prove effective primarily against more blatant forms of deception or when provided with intricate architectural insights about the model under scrutiny.

Nonetheless, as underscored by our typology, numerous untapped research opportunities beckon, aimed at fortifying the prevention of AI misuse and the promotion of ethical AI deployment.

Author Contributions Third author: Proofreading, editing, suggestion for future work, careful checking, and feedback for theory. Second author: Contributed to typology(2.1) /explainee model (2.2), introduction, and related work.First author: Did most of the work.

Funding Open access funding provided by University of Liechtenstein. No grants. Internal funding.

Availability of Data and Materials Data from experiments is public and referenced.

Code Availability All models are public. Additional code can be provided.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval The published conference version was done according to the university's ethical standards. Since the journal version does not include additional data analysis or experiments involving humans compared to the conference version, no explicit approval was sought.

Consent to Participate The authors declare that they have no conflict of interest.

Consent for Publication Not applicable.

References

1. Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). IEEE Access. 2018;6:52138–60.
2. Adebayo J, Gilmer J, Muelly M, et al. Sanity checks for saliency maps. In: Neural information processing systems 2018.
3. Adelani D, Mai H, Fang F, et al. Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection 2019. arXiv:1907.09177
4. Aivodji U, Arai H, Fortineau O, et al. Fairwashing: the risk of rationalization. In: Int. Conf. on Machine Learning(ICML) 2019.
5. Aroyo AM, Gonzalez-Billandon J, Tonelli A, et al. Can a humanoid robot spot a liar? In: Int. Conf. on Humanoid Robots, 2018;1045–1052
6. Bach S, Binder A, Montavon G, et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE. 2015;10: e0130140.
7. Banovic N, Yang Z, Ramesh A, et al. Being trustworthy is not enough: how untrustworthy artificial intelligence (AI) /can deceive the end-users and gain their trust/. Proc ACM Human-Computer Interact. 2023;7(1):1–17.
8. Bertrand A, Belloum R, Eagan JR, et al. How cognitive biases affect XAI-assisted decision-making: a systematic review. In: Proceedings of the 2022 AAAI/ACM conference on AI, ethics, and society, 2022;78–91.
9. Bond CF Jr, DePaulo BM. Accuracy of deception judgments. Persona Soc Psychol Rev. 2006;10(3):214–34.
10. Chakraborty A, Alam M, Dey V, et al. A survey on adversarial attacks and defences. CAAI Trans Intell Technol. 2021;6(1):25–45.
11. Damer TE. Attacking faulty reasoning. Boston, Massachusetts: Cengage Learning; 2013.
12. DePaulo PJ, DePaulo BM. Can deception by salespersons and customers be detected through nonverbal behavioral cues? J Appl Soc Psychol. 1989;19(18):1552–77.
13. Dictionary (2020) In: Merriam Webster.com, https://www.merriam-webster.com/dictionary/explain, Accessed 14 Jan 2020
14. Dimanov B, Bhatt U, Jamnik M, et al. You shouldn't trust me: learning models which conceal unfairness from multiple explanation methods. In: SafeAI@ AAAI 2020.
15. Ekman P, Friesen WV. Nonverbal leakage and clues to deception. Psychiatry. 1969;32(1):88–106.
16. Fukuchi K, Hara S, Maehara T. Faking fairness via stealthily biased sampling. In: Pro. of the AAAI conference on artificial intelligence 2020.
17. Fusco F, Vlachos M, Vasileiadis V, et al. Reconet: an interpretable neural architecture for recommender systems. In: Proceedings of the 28th international joint conference on artificial intelligence, 2019;2343–2349.
18. Giorgi S, Markowitz DM, Soni N, et al. I slept like a baby: using human traits to characterize deceptive ChatGPT and human text. In: International workshop on implicit author characterization from texts for search and retrieval (IACT'23) 2023.
19. Gregor S, Benbasat I. Explanations from intelligent systems: theoretical foundations and implications for practice. MIS Q 1999;23:497–530.
20. Hauch V, Sporer SL, Michael SW, et al. Does training improve the detection of deception? a meta-analysis. Commun Res. 2016;43(3):283–343.
21. Heo J, Joo S, Moon T. Fooling neural network interpretations via adversarial model manipulation. Adv Neural Inf Process Syst 2019;32. https://proceedings.neurips.cc/paper/2019/hash/7fea637fd6d02b8f0adf6f7dc36aed93-Abstract.html
22. Kim Y. Convolutional neural networks for sentence classification. In: Proc. empirical methods in natural language processing (EMNLP) 2014.
23. Kowsari K, Brown DE, Heidarysafa M, et al. Hdltex: Hierarchical deep learning for text classification. In: IEEE Int. conference on machine learning and applications (ICMLA) 2017.
24. Krishna S, Han T, Gu A, et al. The disagreement problem in explainable machine learning: a practitioner's perspective 2022. arXiv preprint arXiv:2202.01602
25. Lai V, Tan C. On human predictions with explanations and predictions of machine learning models: a case study on deception detection. In: Proceedings of the conference on fairness, accountability, and transparency, 2019;29–38.
26. Lakkaraju H, Bastani O. How do I fool you? Manipulating User Trust via Misleading Black Box Explanations. In: Proceedings of the AAAI/ACM conference on AI, ethics, and society, 2020;79–85.
27. Lertvittayakumjorn P, Toni F. Human-grounded evaluations of explanation methods for text classification 2019. arXiv preprint arXiv:1908.11355
28. Levine TR. Encyclopedia of deception. Sage Publications; 2014.

29. Ludwig S, Van Laer T, De Ruyter K, et al. Untangling a web of lies: exploring automated detection of deception in computer-mediated communication. J Manag Inf Syst. 2016;33(2):511–41.
30. Maas A, Daly R, Pham P, et al. Learning word vectors for sentiment analysis. In: Association for computat. linguistics (ACL) 2011.
31. Masip J. Deception detection: state of the art and future prospects. Psicothema. 2017;29:149–59.
32. Mehrabi N, Morstatter F, Saxena N, et al. A survey on bias and fairness in machine learning 2019. arXiv preprint arXiv:1908.09635
33. Meske C, Bunde E, Schneider J, et al. Explainable artificial intelligence: objectives, stakeholders, and future research opportunities. Inf Syst Manag. 2022;39:53–63.
34. Mohseni S, Zarei N, Ragan ED. A multidisciplinary survey and framework for design and evaluation of explainable AI systems. Trans Interact Intell Syst. 2021;11:1–45.
35. Nourani M, Kabir S, Mohseni S, et al. The effects of meaningful and meaningless explanations on trust and perceived system accuracy in intelligent systems. In: AAAI conference on artificial intelligence 2019.
36. Pandey AV, Rall K, Satterthwaite ML, et al. How deceptive are deceptive visualizations? An empirical analysis of common distortion techniques. In: Proceedings of the 33rd annual acm conference on human factors in computing systems, 2015;1469–1478.
37. Papenmeier A, Englebienne G, Seifert C. How model accuracy and explanation fidelity influence user trust 2019. arXiv preprint arXiv:1907.12652
38. Pérez-Rosas V, Kleinberg B, Lefevre A, et al. Automatic detection of fake news 2017. arXiv preprint arXiv:1708.07104
39. Petsiuk V, Das A, Saenko K. Rise: Randomized input sampling for explanation of black-box models 2018. arXiv preprint arXiv:1806.07421
40. Prabhakaran V, Hutchinson B, Mitchell M. Perturbation sensitivity analysis to detect unintended model biases 2019. arXiv preprint arXiv:1910.04210
41. Przybyla P. Capturing the style of fake news. In: Proceedings of the AAAI conference on artificial intelligence, 2020;490–497.
42. Ross AS, Hughes MC, Doshi-Velez F. Right for the right reasons: training differentiable models by constraining their explanations. In: Int. joint conference on artificial intelligence (IJCAI) 2017.
43. Schlegel R, Obermeier S, Schneider J. Structured system threat modeling and mitigation analysis for industrial automation systems. In: International conference on industrial informatics 2017.
44. Schneider J. Foundation models in brief: A historical, socio-technical focus 2022. arXiv preprint arXiv:2212.08967
45. Schneider J, Apruzzese G. Concept-based adversarial attacks: Tricking humans and classifiers alike. IEEE symposium on security and privacy (S &P) workshop on deep learning and security 2022.
46. Schneider J, Apruzzese G. Dual adversarial attacks: fooling humans and classifiers. J Inf Secur Appl. 2023;75: 103502.
47. Schneider J, Breitinger F. Towards AI forensics: did the artificial intelligence system do it? J Inf Secur Appl. 2023;76(103):517.
48. Schneider J, Handali JP. Personalized explanation for machine learning: a conceptualization. In: European conference on information systems (ECIS) 2019.
49. Schneider J, Vlachos M. Personalization of deep learning. In: Data science–analytics and applications 2021.
50. Schneider J, Vlachos M. Explaining classifiers by constructing familiar concepts. Mach Learn 2022;112:1–34.
51. Schneider J, Vlachos M. Reflective-net: Learning from explanations. Data Min Knowl Discov 2023;1–22. https://doi.org/10.1007/s10618-023-00920-0
52. Schneider J, Basalla M, vom Brocke J. Creativity of deep learning: conceptualization and assessment. In: International conference on agents and artificial intelligence (ICAART) 2022.
53. Schneider J, Meske C, Vlachos M. Deceptive AI explanations: Creation and detection. In: Proceedings of the 14th International conference on agents and Artificial intelligence - Volume 2: ICAART,, 2022;44–55.
54. Schwalbe G, Finzel B. A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. Data Min Knowl Discov 2023;1–59. https://doi.org/10.1007/s10618-022-00867-8
55. Selvaraju RR, Cogswell M, Das A, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Int. conference on computer vision (ICCV) 2017.
56. Sison AJG, Daza MT, Gozalo-Brizuela R, et al. ChatGPT: More than a weapon of mass deception, ethical challenges and responses from the human-Centered artificial intelligence (HCAI) perspective 2023. arXiv preprint arXiv:2304.11215
57. Slack D, Hilgard S, Jia E, et al. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In: AAAI/ACM conference on AI, ethics, and society 2020.
58. Van Eemeren FH, Garssen B, Meuffels B. Fallacies and judgments of reasonableness: empirical research concerning the pragma-dialectical discussion rules, vol. 16. Dordrecht: Springer Science & Business Media; 2009.
59. Viering T, Wang Z, Loog M, et al. How to manipulate cnns to make them lie: the gradcam case 2019. arXiv preprint arXiv:1907.10901
60. Wilking R, Jakobs M, Morik K. Fooling Perturbation-Based Explainability Methods. In: Workshop on trustworthy artificial intelligence as a part of the ECML/PKDD 22 program 2022.
61. Wölker A, Powell TE. Algorithms in the newsroom? news readers' perceived credibility and selection of automated journalism. Journalism 2018.
62. Wu Y, Ngai EW, Wu P, et al. Fake online reviews: Literature review, synthesis, and directions for future research. Decis Support Syst. 2020;132: 113280.
63. Wu Y, Ma Y, Wan S. Multi-scale relation reasoning for multi-modal visual question answering. Signal Process : Image Commun. 2021;96(116):319.
64. Xiao B, Benbasat I. Product-related deception in ecommerce: a theoretical perspective. MIS Q. 2011;35(1):169–95.
65. Zhang Q, Wang W, Zhu SC. Examining cnn representations with respect to dataset bias. In: AAAI Conf. on artificial intelligence 2018.