Deceptive AI Explanations: Creation and Detection

Johannes Schneider[1*], **Joshua Peter Handali**[1], **Michalis Vlachos**[2], **Christian Meske**[3]
[1]University of Liechtenstein, Vaduz, Liechtenstein
[2]University of Lausanne, Lausanne, Switzerland
[3]Free University of Berlin, Berlin, Germany
{johannes.schneider, joshua.handali}@uni.li, {michalis.vlachos}@unil.ch,
{christian.meske}@fu-berlin.de

Abstract

Artificial intelligence comes with great opportunities and but also great risks. We investigate to what extent deep learning can be used to create and detect deceptive explanations that either aim to lure a human into believing a decision that is not truthful to the model or provide reasoning that is non-faithful to the decision. Our theoretical insights show some limits of deception and detection in the absence of domain knowledge. For empirical evaluation, we focus on text classification. To create deceptive explanations, we alter explanations originating from GradCAM, a state-of-art technique for creating explanations in neural networks. We evaluate the effectiveness of deceptive explanations on 200 participants. Our findings indicate that deceptive explanations can indeed fool humans. Our classifier can detect even seemingly minor attempts of deception with accuracy that exceeds 80% given sufficient domain knowledge encoded in the form of training data.

1 Introduction

Because of the limited moderation of online content, attempts at deception proliferate. Online media struggle against the plague of "fake news", and e-commerce sites spend considerable effort in detecting deceptive product reviews. For example, marketing strategies exist that consider the creation of fake reviews to make products appear better or to provide false claims about product quality [Adelani *et al.*, 2019].

One can consider other examples of why to provide "altered" explanations of a predictive system. Explanations might allow to re-engineer the logic of the AI system, i.e., leak intellectual property. Decision makers might also deviate from suggested AI decisions at will, eg. a bank employee might deny a loan to a person she dislikes claiming an AI model's recommendation as reason (irrespective of the actual recommendation of the system) with a made-up explanation. The learning process of AI systems might yield systems with better performance when utilizing information that should not be used but is available. For example, basing hiring decisions on gender or ethnicity is forbidden in many countries, but using it might yield better job performance predictions of applicants. As such there may be an incentive to build systems that utilize such information, but hide its use. That is, "illegal" decision criteria are used but they are omitted from explanations requested by authorities or even citizens. In Europe, the GDPR law grants such rights to individuals for decisions made in an automated manner.

Frequently, both the decision and the explanation are generated automatically. However, generating explanations is far from trivial. AI systems, commonly based on deep learning, are often very complex consisting of millions of neurons. Still, in recent years there has been tremendous effort in creating methods for improving transparency of such "black boxes" via explanations [Schneider and Handali, 2019]. Evaluations in the context of textual explanation [Lertvittayakumjorn and Toni, 2019] show that automatically generated explanations are deemed helpful for certain tasks. Thus, given economic incentives to provide incorrect explanations, the question arises, *whether humans can be deceived by automatically generated explanations and to what extent deceptive explanations can be detected*.

In this work, we are interested in the automatic generation of explanations supporting deception either by providing incorrect explanations for a correct decision or by generating an explanation for an incorrect decision. We focus on the case, where a recipient might be able to detect a lie by critically examining an explanation, the decision and the information used for decision-making. For illustration, for product reviews, a person may be lured to believe that a review is positive despite it being negative. To this end, the person might be shown the entire review, but seemingly positive terms could be highlighted to give the impression of being positive, while negative terms are left unremarkable.

2 Problem Statement

A machine learning (ML) model M maps an input $X \in S$ to an output Y, where S is the set of all possible inputs. An explainee (the recipient of an explanation) obtains for an input X, a decision D and an explanation H that is allegedly capturing the model behavior. The decision D is claimed to be the model's prediction $Y = M(X)$. The goal of a deceiver is to construct an explanation so that the explainee is neither suspicious about the decision in case it is not truthful

*Contact Author

Explanation \ Reported Prediction	True to model, D(X)=M(X)	Unfaithful, D(X)≠M(X)
True to model	(TT) Telling the truth	(FT) Altered prediction with supporting explanation
Unfaithful	(TT) Non-altered prediction with incorrect explanation	(FT) Altered prediction with incorrect explanation

Figure 1: Scenarios for reported predictions and explanations

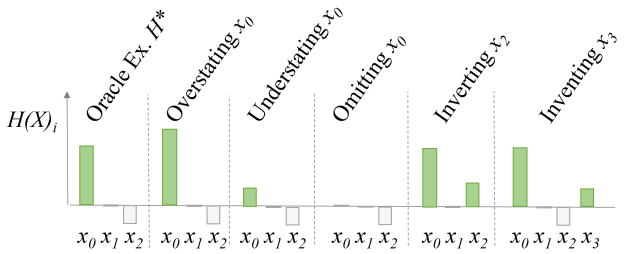

Figure 2: Deviations from ground truth, ie. oracle explanation H^*.

to the model ($D \neq M(X)$) nor about the explanation H, if it deviates from the ground-truth explanation ($H \neq H^*$) that precisely provides the reasoning of the model (see Figure 1).

An input X consists of values for n features, $\mathcal{F} = \{i | i = 1 \dots n\}$, 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, 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. The reported decision $D(X)$ might differ from the model decision $Y = M(X)$. We assume that there is an oracle that provides ground-truth explanations H^* describing why a model M would output Y' for input X, ie. $H^*_M(X, Y')$. We omit M (if clear in a context) and assume $Y' = D(X)$, if not stated otherwise, ie. write $H^*(X)$. Note that the oracle can provide explanations for a hypothetical decision $D(X) \neq M(X)$, ie. "Given that the model decided $D(X) \neq M(X)$, what would have made it do so?". To this date, there might not be an oracle that outputs a "perfect" explanation according to an individual's judgement [Schneider and Handali, 2019]. Thus, we use a proxy method that aims to give truthful explanations and behaves in a known, predictable manner such as LIME [Ribeiro *et al.*, 2016] or Grad-CAM [Selvaraju *et al.*, 2017]. While ML models might learn a hierarchy of features, explaining in terms of learnt features is challenging, since they might not be mapped easily to features or 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 $H(X)$ that output a value $H(X)_i$ for each feature $i \in F$. Where $H(X)_i > 0$ implies that feature i with value x_i is supportive of decision $D(X)$. A value of zero implies no dependence of i on the decision Y. $H(X)_i < 0$ shows that another decision is supported.

2.1 Explanation Faithfulness

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

Decision fidelity. It amounts to the standard notion of quantifying whether input X and explanation $H(X, D(X))$ on their own allow to derive the model decision $Y = M(X)$ [Schneider and Handali, 2019]. Therefore, if reported decisions $D(X)$ differ from model decisions $D(X) \neq M(x)$ or explanations are indicative of multiple outputs, this is hardly possible. Decision fidelity f_D can be defined as the loss when predicting the outcome using some classifier g based on the explanation only, or formally:

$$f_D(X) = -L(g(H, X), Y) \tag{1}$$

The loss might be defined as 0 if $g(H, X) = Y$ and 1 otherwise. We assume that the oracle explanations H^* result in minimum loss, ie. maximum decision fidelity. (Large) decision fidelity does not require that an explanation contains all relevant features used to derive the decision D. 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.

Oracle fidelity. This refers to the overlap of a ground-truth explanation of an oracle H^* with the (potentially deceptive) explanation H for an input X and reported decision $D(X)$. Any mismatch of a feature in the explanation H and H^* lowers oracle fidelity. It is defined as:

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

Note, that even if the decision $D(X)$ is non-truthful to the model, ie. $D(X) \neq M(X)$, oracle fidelity might be large, if the explanation correctly outputs the reasoning that would lead to the reported decision. In the case that the reported decision is truthful, ie. $D(X) = M(X)$, there seems to be an obvious correlation between decision- and oracle fidelity. But any arbitrarily small deviation of oracle fidelity from the maximum of 1 does not necessarily ensure large decision fidelity and vice versa. For example, assume that the explanation H systematically under- or overstates the relevance of features, ie. $H(X)_i = H(X)_i \cdot c_i$ 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 oracle fidelity. However, decision fidelity might yield the opposite picture, ie. maximum decision fidelity, since a classifier g (Def. 1) trained on inputs (X, H) with labels $D(X)$, might learn the coefficients c_i and predict labels without errors.

Oracle fidelity captures the degree of deceptiveness of explanation H by aggregating the differences of its relevances of features and those of a hypothesized perfect explanation H^*. 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 Figure 2). In this work, we do not consider fabricating by (input) feature invention. Omission and inverting of features can be viewed as special cases of over- and understating.

2.2 Purposes of Deceptive Explanation

Some purposes of a deceptive explanation are:

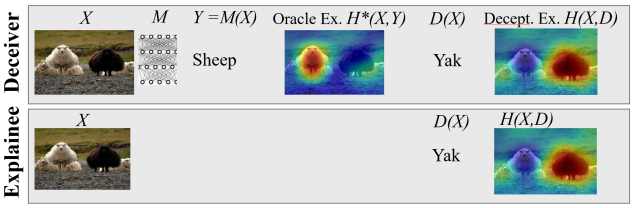

Figure 3: Inputs and outputs for deceiver and explainee for scenario FT in Figure 1. Images by [Petsiuk *et al.*, 2018].

(i) Convincing the explainee of an incorrect prediction, ie. that a model decided D for an input X although the model's output is $Y = M(X)$ with $Y \neq D$.

(ii) Providing an explanation that does not accurately capture model behavior without creating suspicion. An incorrect explanation will manifest in low decision fidelity and oracle fidelity. It involves hiding or overstating the importance of features in the decision process (Figure 2) 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 in 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 Figure 1.

To construct deceptive explanations (and decisions), a deceiver has access to the model M, the input X and outputs a decision D in combination with an explanation $H(X)$ (see Figure 3). Deceptive explanations are constructed to maximize the explainee's credence of decisions and explanations. We assume that an explainee is most confident that an oracle explanation $H^*(X, Y)$ and the model based decision $Y = M(X)$ are correct. The explainee might rely not just on a single explanation and decision for one input $X \in S$, but reason using explanations and decisions of multiple inputs $S' \subseteq S$.

2.3 Detection Problem

We consider two detection problems:

Max-credence Problem. The goal is to maximize the percentage of decisions and explanations that are identified correctly either as truthful (TT in Figure 1) or otherwise.

Concept Fidelity Problem. The goal is to compute the expected oracle fidelity of a feature value v:

$$f_O(v) := \sum_{i,X \in \mathcal{F}(v,S)} \frac{f_O(X)_i}{|\mathcal{F}(v,S)|} \quad \text{with} \tag{3}$$

$$\mathcal{F}(v,T) := \{(j,X) | x_j = v, X = (x_i), X \in T\} \tag{4}$$

$\mathcal{F}(v,T)$ denotes the set of all inputs in set $T \subseteq S$ and their features with value v. Concept fidelity allows, for example, to check if features are omitted. Say that values $V' \subseteq V_i$ are related to gender or race, eg. $V' = \{male, female\}$. To assess if values V' impact decisions, explainees have to assess that they are not reported as relevant in explanations, ie. $H(X)_i = 0$ for all inputs $(i, X) \in \mathcal{F}(v, T)$ an all $v \in V'$, and that they have maximum oracle fidelity, ie. $f_O(X)_i = 1$.

3 Classification and Explanation

We elaborate on two text classification tasks using a convolutional neural network (CNN) for text classification by Kim [2014] and GradCAM [Selvaraju *et al.*, 2017] for generating oracle explanations. 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 [Adebayo *et al.*, 2018]. While GradCAM is most commonly employed for CNN on image recognition the mechanisms for texts are identical. In fact, [Lertvittayakumjorn and Toni, 2019] showed that GradCAM on CNNs similar to the one by Kim [2014] leads to outcomes on human tasks that are comparable to other explanation methods such as LIME.

The GradCAM method, which serves as our oracle explanation H^*, computes a gradient-weighted activation map starting from a given layer or neuron within that layer back to the input X. To obtain an oracle explanation $H^*(X, Y')$, we use the neuron before the softmax layer that represents the class Y' to explain. For generating a high fidelity explanation for an incorrectly reported predictions $D(X) \neq M(X)$ (scenario FT in Figure 1) we provide as explanation the oracle explanation, ie. $H(X, D(X)) = H^*(X, D(X))$. By definition oracle explanations maximize oracle fidelity f_O.

For scenarios involving non-truthful explanations (TF and FF) 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 oracle 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:*

$$H_{Omit}(X)_i := \begin{cases} 0, & \text{if } x_i \in \mathcal{V}. \\ H^*(X)_i, & \text{otherwise}. \end{cases} \tag{5}$$

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, eg. to prevent re-engineering through obfuscation.

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

$$H_{Noise}(X)_i := H^*(X)_i \cdot (1 + r_{i,X}), \tag{6}$$

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

4 Detection

Ideally, any of the four types of deception (see Figure 1) are detected using only one or more inputs $S' \subseteq S$ and their responses $D(X)$ and $H(X)$ for $X \in S'$ (see Figure 3). But, without additional domain knowledge or context information, it is impossible to distinguish among the four scenarios 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, eg. 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.

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

Proof. Consider a model M for dataset $\{(X, Y)\}$ for binary classification with class labels $Y \in \{0, 1\}$ with $D(X) = M(X)$. Assume a deceiver switches the decision of model M, ie. it returns $D(X) = 1 - M(X)$ and $H^*(X, M(X), M)$. Consider a dataset with switched labels, ie. $\{(X, 1 - Y)\}$ and a second model M' that is identical to M except that it outputs $M'(X) = 1 - Y = 1 - M(X)$. Thus, oracle explanation are identical, ie. we have $H^*(X, M'(X), M') = H^*(X, M(X), M)$. Thus, for input X both the deceiver and model M' report $D(X) = 1 - M(X)$ and $H^*(X, M(X), M)$. Therefore, they cannot be distinguished by any detector. □

A similar theorem might be stated for non-truthful explanations $H \neq H^*$, eg. by using feature inversion $H(X) = -H^*(X)$.

Next, we provide methods to detect some forms of deceptions without and with domain knowledge, ie. human labeled training data. More precisely, for the max-credence problem we use a supervised learning approach. That is, we are given a training set, where each sample consists of the three inputs ($H(X, D)$, X, D) together with label $L \in \{TT, FT, TF, FF\}$ stating the scenario in Figure 1. Thus, our goal is to develop a classifier maximizing deception detection accuracy.

For the concept fidelity problem, computing $f_O(v)$ is difficult, since it requires oracle explanations H^*. Therefore, we aim at an approximate, surrogate measure that might only help in identifying features that are possibly misrepresented in the explanations. Our idea is partially motivated by the following theorem stating 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 $D(X) \neq D(X')$ 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 compute for each input $X \in S$, the prediction $D(X)$ and explanation $H(X)$. By Definition 1, if feature values $\mathcal{V}$ are omitted it must hold $H(X)_i = 0$ 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 $M(X) \neq M(X')$. The latter holds because the change in decision must be attributed to a non-zero relevance of $x_j \in \mathcal{V}$ or $x'_j \in \mathcal{V}$, since X and X' are identical except for feature j with values that are deemed omitted. □

The proof of Theorem 2 is constructive, but generally all possible 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. Therefore, we rely on the idea that if a feature i, ie. its value, is reported to be highly relevant, ie. $H(X)_i$ is large, then removing, adding or changing a feature i with value $x_i = v$ should more likely result in a change of prediction than feature values that are deemed less relevant. Computation is related to the Shapley value. For our text classification, we define the average explanation relevance for a specific value, ie. word v, for a subset of inputs $S' \subseteq S$ as follows:

$$\overline{H}(v, S') := \frac{\sum_{(j,X') \in \mathcal{F}(v,S')} |H(X')_j|}{|\mathcal{F}(v, S')|} \tag{7}$$

Addition of a word v to samples $S'_{v \notin S'}$ that do not contain word v is done by replacing a random feature value with v. Let X_{+v} be the sample that results from changing $X \in S'_{v \notin S'}$ by such a replacement. We compute the accuracy $A_+(v)$ due to adding v by replacement, ie. how many predictions $M(X)$ and $M(X_{+v})$ match for $X \in S'_{v \notin S'}$.[1]

We use the accuracy $A_+(v)$ and the average explanation relevance $\overline{H}(v, S')$ to coarsely estimate $f_O(v)$. The more likely a point $(A_+(v), \overline{H}(S'_v))$ is an outlier, the larger $f_O(v)$ is expected to be.

5 Empirical Investigation

We evaluate generation of deceptive explanations and their detection. For detection, we conduct a user study and also use ML models. We assume that only features that are claimed to contribute positively to a decision are included in explanations. That is, features that are claimed to be irrelevant or even supporting of another possible decision outcome are ignored. The motivation is that we aim at explanations that are as simple for a human to understand as possible.

5.1 Setup

We employed two datasets. The IMDB dataset [Maas *et al.*, 2011] consisting of movie reviews and a label indicating sentiment polarity, ie. either positive or negative. We also utilized the Web of Science (WoS) dataset consisting of abstracts of scientific papers classified into 7 categories (Psychology, Medical Science, Civil Engineering, Mechanical Engineering, Computer Science and Electrical Engineering) [Kowsari *et al.*, 2017]. Our CNNs for classification achieved accuracies of 87% for IMDB and 75% for WoS trained using the AdamOptimizer for 300 epochs with 2/3 of the samples for training and 1/3 for testing. We computed explanations for test data only.

5.2 Human-based Detection

We conducted a user study using the IMDB dataset (Section 5.1).[2] For the scenarios of interest, we compare explanations which are aligned to the shown prediction, ie. TT and FT. Two samples are shown in Figure 4.

[1] Removal is analogous. It is not considered due to space.

[2] 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 [Lertvittayakumjorn and Toni, 2019]

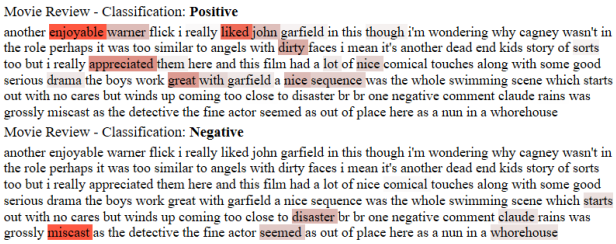

Figure 4: Generated sample explanations for scenarios TT (top) and FT (bottom) from Figure 1

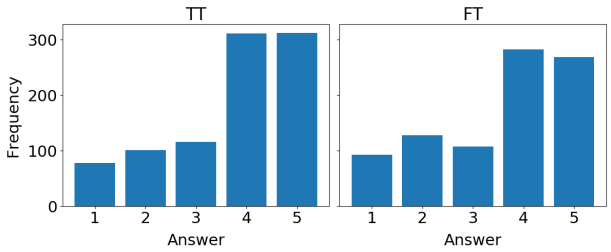

Figure 5: Distributions of user replies to "The classification is correct" (1 = strongly disagree to 5 = strongly agree).

Participants. 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 25 predictions together with explanations for each dataset. We randomized the choice of presented samples, ie. we randomly chose a dataset, randomly chose a sample of the dataset, randomly chose between scenarios TT and FT in Figure 1.

5.3 Machine Learning-based Detection

For the detection experiment, we chose samples that were predicted correctly by our trained classifier. We investigate detection of deceptive explanations using several methods on all four scenarios in Figure 1 under the following criteria. We omitted a randomly chosen set of words V (see Def. 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 used both 0.01 and 0.25 (both noise and omission). Since labeling is difficult and potentially error-prone, we consider different levels of label noise, ie. $L \in [0, 0.32]$ such that a fraction L of all labels were replaced with a random label (different from the correct one).

Models. For the Random Forest model we chose 1000 trees of depth 12. The model input is a concatenation of three vectors: i) a textvector of word indices, ii) a heatmap vector of values obtained via GradCAM, that is a 1:1 mapping of the visual output shown to the user, and iii) a one-hot prediction vector of the decision. The CNN classifier is designed as follows: we modified the CNN for text classification by concatenating the heatmap vector with the word embedding before the convolution layer. The one-hot prediction vector is then concatenated with the output of the convolution layer.

5.4 Heuristic-based Detection

The text classifier considered the 20,000 most common words. We randomly selected 3,000 out of the 20,000 words to investigate their impact on the decision if they are omitted and added.

6 Results

6.1 Human-based Detection

Out of the 200 participants, we removed participants that spend less than 5 seconds 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 133 participants amounting to 3,500 answers shown in Figure 5 for both scenarios.

A t-test of means confirmed that the distributions differ significantly (p-value of 0.008), though the mean scores 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 often times, they have the capability to collectively detect deceptive explanations.

6.2 Machine Learning-based Detection

Figure 6 suggests that without label noise, under all scenarios, accuracy exceeds 90% for correctly deciding whether a decision and/or explanation are deceptive. Accuracy worsens under greater label noise though even for a noise level of 32% it is better than chance. The more extreme the distortion of explanations, ie. the larger k, the higher detection accuracy. But even for minor modifications, ie. 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 the scenario TFNoise (in Figure 6), ie. 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. For larger noise levels ($k = 0.25$), random forests perform much better. CNNs also do well for $k = 1$, where test accuracies exceed 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 $D(X)$ without much need for an explanation. This is supported by the observation that the degree of deception manifested in the explanations has 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.

6.3 Heuristic-based Detection

The scatter plots in Figure 7 highlight that feature values v with larger relevance $\overline{H}$ have stronger impact on the decision. All explanations are truthful. Any point, ie. word, that is outside the dense area in any of the plots is a potential candidate for not being truthfully represented in the explanations. For example, the fictitious point $(\overline{H}, A_+) = (0.2, 0.85)$ for the

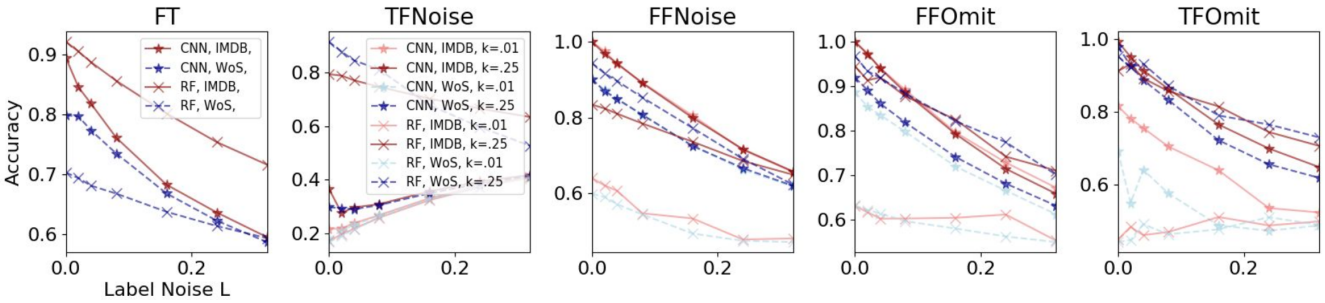

Figure 6: ML-based detection results for scenarios in Figure 1

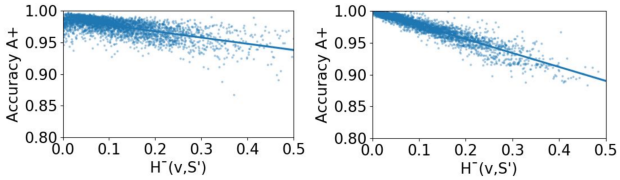

Figure 7: Plot of WoS data(left) and IMDB(right)

IMDB dataset poses an outlier that might well stem from deceptive explanations. Thus, while deceptions of feature values leading to large impact on accuracy might be well visible, those that have only small impact on accuracy might not be.

7 Related Work

[Viering *et al.*, 2019] are interested in manipulating the inner workings of a CNN, so that it outputs arbitrary explanations. Whether the explanations themselves are convincing or not, is not considered, ie. the paper shows many examples of "incredible" explanations that can easily be detected as non-genuine. Lakkaraju and Bastani [2019] investigated the effectiveness of misleading explanations to manipulate user's trust. Decisions were made using prohibited features such as gender and race but misleading explanations were supposed to disguise their usage. Their study found that users can be manipulated into trusting high fidelity but misleading explanations for correct predictions. Their setup pertains to one of our four scenarios, ie. FT. In contrast to this work, they have a more domain expert-oriented and problem specific focus. They lack automatic detection and the concept of oracle fidelity. Papenmeier et al. [2019] investigated the influence of classifier accuracy and explanation fidelity on user trust. For three classifiers (differing strongly in test accuracy), they considered "random" explanations, ie. using randomly chosen features, and "oracle" explanations, ie. explanation made by an automatic method. They found that accuracy is more relevant for trust than explanation quality though both matter.

[Nourani *et al.*, 2019] investigated upon the impact of explanations on trust. Poor explanations indeed reduce a user's perceived accuracy of the model, independent of its actual accuracy. Explanations' helpfulness varies depending on task and method [Lertvittayakumjorn and Toni, 2019]. Explanations are more helpful in assessing a model's predictions compared to its behaviour. 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 [Bach *et al.*, 2015] helps in assessing uncertain predictions.

[Adelani *et al.*, 2019] 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[Pérez-Rosas *et al.*, 2017] based on ML methods and linguistic features obtained through dictionaries. Linguistic cues [Ludwig *et al.*, 2016] such as flattery was used to detect deception in e-mail communication. We do not encode explicit, domain-specific detection features such as flattery.

Our methods might be valuable for the detection of fairness and bias – see [Mehrabi *et al.*, 2019] for a recent overview. There are attempts to prevent machine learning techniques from making decisions based on certain attributes in the data, such as gender or race [Ross *et al.*, 2017] or to detect learnt biases based on representations [Zhang *et al.*, 2018] or perturbation analysis for social associations [Prabhakaran *et al.*, 2019]. In our case, direct access to the decision-making system is not possible — neither during training nor during operations, but we utilize explanations.

In human-to-human interaction behavioral cues such as response times [Levine, 2014] or non-verbal leakage due to facial expressions [Ekman and Friesen, 1969] might have some, but arguably limited impact [Masip, 2017] on deception detection. In our context, this might pertain, eg. to computation time. We do not use such information. Explanations to support deceptions typically suffer from at least one fallacy such as "the use of invalid or otherwise faulty reasoning" [Van Eemeren *et al.*, 2009]. Humans can use numerous techniques to attack fallacies [Damer, 2013], often based on logical reasoning. Such techniques might also be valuable in our context. In particular, ML techniques have been used to detect lies in human-interaction, eg. [Aroyo *et al.*, 2018].

8 Conclusions

Given economic and other incentives, we believe that another cat and mouse game between "liars" and "detectors" will emerge in the context of AI. Our work provided a first move in this game: We contributed by showing that detection of deception attempts without domain knowledge is not

always possible. But machine learning models utilizing domain knowledge through training data yield good detection accuracy.

References

[Adebayo *et al.*, 2018] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *NIPS*, pages 9505–9515, 2018.

[Adelani *et al.*, 2019] David Adelani, Haotian Mai, Fuming Fang, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection. *arXiv:1907.09177*, 2019.

[Aroyo *et al.*, 2018] Alexander Mois Aroyo, J Gonzalez-Billandon, A Tonelli, Alessandra Sciutti, Monica Gori, Giulio Sandini, and Francesco Rea. Can a humanoid robot spot a liar? In *Int. Conf. on Humanoid Robots*, pages 1045–1052, 2018.

[Bach *et al.*, 2015] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[Damer, 2013] T Edward Damer. *Attacking faulty reasoning*. Cengage Learning, Boston, Massachusetts, 2013.

[Ekman and Friesen, 1969] Paul Ekman and Wallace V Friesen. Nonverbal leakage and clues to deception. *Psychiatry*, 32(1):88–106, 1969.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *Proc. EMNLP*, pages 1746–1751, 2014.

[Kowsari *et al.*, 2017] Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S Gerber, and Laura E Barnes. Hdltex: Hierarchical deep learning for text classification. In *IEEE ICMLA*, pages 364–371, 2017.

[Lakkaraju and Bastani, 2019] Himabindu Lakkaraju and Osbert Bastani. How do i fool you?: Manipulating user trust via misleading black box explanations. *arXiv preprint arXiv:1911.06473*, 2019.

[Lertvittayakumjorn and Toni, 2019] Piyawat Lertvittayakumjorn and Francesca Toni. Human-grounded evaluations of explanation methods for text classification. *arXiv preprint arXiv:1908.11355*, 2019.

[Levine, 2014] Timothy R Levine. *Encyclopedia of deception*. Sage Publications, 2014.

[Ludwig *et al.*, 2016] Stephan Ludwig, Tom Van Laer, Ko De Ruyter, and Mike Friedman. Untangling a web of lies: Exploring automated detection of deception in computer-mediated communication. *Journal of Management Information Systems*, 33(2):511–541, 2016.

[Maas *et al.*, 2011] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proc. ACL*, pages 142–150, 2011.

[Masip, 2017] Jaume Masip. Deception detection: State of the art and future prospects. *Psicothema*, 29(2):149–159, 2017.

[Mehrabi *et al.*, 2019] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

[Nourani *et al.*, 2019] Mahsan Nourani, Samia Kabir, Sina Mohseni, and Eric D Ragan. The effects of meaningful and meaningless explanations on trust and perceived system accuracy in intelligent systems. In *Proc. AAAI*, pages 97–105, 2019.

[Papenmeier *et al.*, 2019] Andrea Papenmeier, Gwenn Englebienne, and Christin Seifert. How model accuracy and explanation fidelity influence user trust. *arXiv preprint arXiv:1907.12652*, 2019.

[Pérez-Rosas *et al.*, 2017] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*, 2017.

[Petsiuk *et al.*, 2018] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[Prabhakaran *et al.*, 2019] Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. Perturbation sensitivity analysis to detect unintended model biases. *arXiv preprint arXiv:1910.04210*, 2019.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proc. ACM SIGKDD*, pages 1135–1144, 2016.

[Ross *et al.*, 2017] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: training differentiable models by constraining their explanations. In *Proc. IJCAI*, pages 2662–2670, 2017.

[Schneider and Handali, 2019] Johannes Schneider and Joshua Peter Handali. Personalized explanation for machine learning: a conceptualization. In *Proc. ECIS*, 2019.

[Selvaraju *et al.*, 2017] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proc. IEEE ICCV*, pages 618–626, 2017.

[Van Eemeren *et al.*, 2009] Frans H Van Eemeren, Bart Garssen, and Bert Meuffels. *Fallacies and judgments of reasonableness: Empirical research concerning the pragma-dialectical discussion rules*, volume 16. Springer Science & Business Media, Dordrecht, 2009.

[Viering *et al.*, 2019] Tom Viering, Ziqi Wang, Marco Loog, and Elmar Eisemann. How to manipulate cnns to

make them lie: the gradcam case. *arXiv preprint arXiv:1907.10901*, 2019.

[Zhang *et al.*, 2018] Quanshi Zhang, Wenguan Wang, and Song-Chun Zhu. Examining cnn representations with respect to dataset bias. In *Proc. AAAI*, 2018.