Computers & Security 120 (2022) 102791

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Tamp-X: Attacking explainable natural language classifiers through tampered activations ☆

Hassan Ali [a,1], Muhammad Suleman Khan [a,1], Ala Al-Fuqaha [b], Junaid Qadir [c,*]

[a] Information Technology University (ITU), Lahore, Pakistan
[b] Information and Computing Technology (ICT) Division, College of Science and Engineering (CSE), Hamad Bin Khalifa University, Doha, Qatar
[c] Department of Computer Science and Engineering, College of Engineering, Qatar University, Doha, Qatar

ARTICLE INFO

Article history:
Received 1 February 2022
Revised 9 May 2022
Accepted 4 June 2022
Available online 6 June 2022

Keywords:
Explainable artificial intelligence (XAI)
Natural language processing
Attacking XAI
Adversarial attacks
Model tampering

ABSTRACT

While the technique of Deep Neural Networks (DNNs) has been instrumental in achieving state-of-the-art results for various Natural Language Processing (NLP) tasks, recent works have shown that the decisions made by DNNs cannot always be trusted. Recently Explainable Artificial Intelligence (XAI) methods have been proposed as a method for increasing DNN's reliability and trustworthiness. These XAI methods are however open to attack and can be manipulated in both white-box (gradient-based) and black-box (perturbation-based) scenarios. Exploring novel techniques to attack and robustify these XAI methods is crucial to fully understand these vulnerabilities. In this work, we propose *Tamp-X*—a novel attack which tampers the activations of robust NLP classifiers forcing the state-of-the-art white-box and black-box XAI methods to generate misrepresented explanations. To the best of our knowledge, in current NLP literature, we are the first to attack both the white-box and the black-box XAI methods simultaneously. We quantify the reliability of explanations based on three different metrics—the descriptive accuracy, the cosine similarity, and the L_p norms of the explanation vectors. Through extensive experimentation, we show that the explanations generated for the tampered classifiers are not reliable, and significantly disagree with those generated for the untampered classifiers despite that the output decisions of tampered and untampered classifiers are almost always the same. Additionally, we study the adversarial robustness of the tampered NLP classifiers, and find out that the tampered classifiers which are harder to explain for the XAI methods, are also harder to attack by the adversarial attackers.

1. Introduction

Deep Neural Networks (DNNs) are increasingly being deployed for various tasks such as healthcare Qayyum et al. (2020), autonomous driving Grigorescu et al. (2020); Khalid et al. (2020), financial applications Ozbayoglu et al. (2020), medical image analysis Petrick et al. (2021), and crime prediction Kounadi et al. (2020). Such a wide use of DNNs is particularly attributed to their impressive performance in solving the real-world problems requiring intelligent decision-making for numerous applications in Computer Vision (CV), Speech Processing and Natural Language Processing (NLP). However, DNNs operate in a black-box fashion which raises concerns regarding the trustworthiness of these models Ribeiro et al. (2016). Recent research has demonstrated that DNNs are vulnerable to attacks at both the training Ali et al. (2020) and the inference stages Khalid et al. (2020) and can exhibit stereotypical bias. This significantly degrades the reliability of DNNs, particularly in settings such as autonomous driving and home IoT devices, where mistakes can harm human beings or be injurious or even fatal.

Recently, in order to enhance the trustworthiness of these models, several "Explainable Artificial Intelligence" (XAI) efforts have focused on explaining the behavior of a Deep Learning (DL) model when provided a particular input or a class of inputs Lundberg and Lee (2017); Ribeiro et al. (2016); Smilkov et al. (2017); Sundararajan et al. (2017). These methods estimate the contribution of input features over the output of a model by either analyzing the gradients of the model (the so-called *white-box methods*) or by observing the effects of the perturbations to an input (and operating as *black-box methods*) Das and Rad (2020). We

☆ This document is the result of the research project funded by the Qatar National Research Fund (a member of Qatar Foundation)
* Corresponding author.
E-mail addresses: hassan.ali@itu.edu.pk (H. Ali), aalfuqaha@hbku.edu.qa (A. Al-Fuqaha), jqadir@qu.edu.qa (J. Qadir).
[1] Hassan Ali and Muhammad Suleman Khan have equal contribution.

https://doi.org/10.1016/j.cose.2022.102791

have seen a growing interest of researchers leveraging these explainability methods to detect/mitigate unintended DNN behaviors such as bias Jain et al. (2020), DNN backdoors Doan et al. (2020), and adversarial attacks Fidel et al. (2020). Recent research has shown that DNNs fail to perform well on adversarially perturbed inputs Ali et al. (2019); Goodfellow et al. (2015). Although a plethora of works exist that exploit XAI methods to counter unreliable DNN behaviors, we note that studying the reliability and fragility of these XAI methods themselves has gained attention only recently Rosenfeld (2021); Warnecke et al. (2020); Yalcin et al. (2021); Zhou et al. (2021).

In order to fool the XAI methods, many adversarial attacks have been proposed, most of which utilize minimal adversarial perturbations to manipulate the gradients and the decision of a model Yeh et al. (2019); Zhang et al. (2020). Although such attacks are effective in the vision domain, the discrete space of Natural Language Processing (NLP) prohibits their convenient use for attacking XAI methods in the NLP domain Ali et al. (2021). This is one of the reasons for significantly limited adversarial research in NLP as compared to that in the field of computer vision.

To the best of our knowledge, the *Scaffolding* attack Slack et al. (2020) and the *FACADE* attack Wang et al. (2020) are the only two attacks in literature that fool XAI methods for NLP classification tasks. However, the Scaffolding attack exploits three different DNNs and only works against the perturbation-based (black-box) XAI methods Slack et al. (2020). On the other hand, the FACADE attack only works for the gradient-based (white-box) XAI methods which generate explanations for a given input by computing local gradients of the NLP classifier for the input. Additionally, the FACADE attack is only partially effective as it cannot fool InteGrad (IG), a gradient-based XAI method, as reported in the original paper Wang et al. (2020).

In this work, we explore a critical question: *Can both the perturbation- and the gradient-based XAI methods be single-handedly fooled by a tampered NLP classifier at the same time?* To answer this question, we propose *Tamp-X*—a novel attack against four popular state-of-the-art explainability techniques—LIME Ribeiro et al. (2016), SHAP Lundberg and Lee (2017), InteGrad (IG) Sundararajan et al. (2017), and SmoothGrad (SG) Smilkov et al. (2017). As an attacker, our goal is to train an NLP classifier such that the explanations generated by XAI methods for inputs to the classifier are incorrect.

In order to achieve our goal, we first train a noise-tolerant classifier *robust* to strong perturbations by randomly masking z words of an input sequence while training. Our motivation for this step stems from the observation that perturbation-based XAI methods compute the feature importance by introducing strong perturbations to the original input. After the model is trained, we *tamper* the activations of the classification layer before applying the softmax activation in order to manipulate the gradients/contributions of input features while keeping the output decision intact. A simplified illustration of our attack can be seen in Fig. 1.

We evaluate explanations generated by the four previously-highlighted XAI methods on three different explanation evaluation metrics—namely, descriptive accuracy Warnecke et al. (2020), cosine similarity, and L_p norms—and empirically show that *Tamp-X* can fool both perturbation- and gradient-based XAI methods. Our experiments motivate the need for new XAI methods, which are robust and not easily manipulable. Additionally, we evaluate our *tampered* classifiers against the state-of-the-art NLP-adversarial attacks using the state-of-the-art *TextAttack* library Morris et al. (2020) and observe that our *tampered* classifiers are difficult to attack as compared to the vanilla classifiers. We attribute this to the obfuscated gradients caused by the tampered activations of our classifiers which gives a false sense of security against the adversarial attacks Athalye et al. (2018). However, unlike the vision domain, where such an obfuscation can be broken using the iterative adversarial attacks Khalid et al. (2020), which is made possible by a continuous input space, we find no attack in current NLP literature which addresses this issue because the input space of NLP is discrete. Finally, we provide important insights for the future researchers to develop mitigation techniques for our proposed attack.

Our major contributions are summarized below,

- To the best of our knowledge, *Tamp-X* is the first attack in the current NLP literature which can fool both the black- and the white-box XAI methods single-handedly. We show that *Tamp-X* can manipulate the generated explanations by craftly tampering the activation functions of robust NLP classifiers without sacrificing the accuracy over the clean inputs.
- We study our tampered and untampered NLP classifiers under the state-of-the-art adversarial attacks and find that the tampered NLP classifiers are significantly harder to attack as compared to the untampered classifiers.
- We observe a trade-off between the explainability of an NLP classifier, and its adversarial robustness. More specifically, in our experiments, tampering NLP classifier to attack the XAI methods makes it harder for an adversarial attacker to launch misclassification attacks. We explain this based on similar formulations of XAI methods and adversarial attacks—both work by first estimating the contributions of input words over the classifier's output.

2. Background and related work

2.1. Explainable artificial intelligence (XAI)

Although a number of XAI methods have been proposed in literature, these techniques can generally be classified into two main categories—white-box and black-box XAI methods. The white-box methods assume complete knowledge—including the architecture, the number of layers and the learned weights—of the classifier under inspection. Such methods generally leverage the classifier's gradients to explain its decision on a particular input. In contrast to the white-box methods, the black-box XAI methods do not assume such knowledge about the underlying classifier which is to be inspected. Such methods usually generate explanations by carefully perturbing the input and measuring the change at the output. In the following, we provide a brief overview along with the notable works in each of these categories.

2.1.1. Black-Box XAI methods

Black-box XAI methods generally calculate the features attribution by perturbing the classifier input, and monitoring the change at the output.

LIME and SHAP are two of the most popular black-box XAI methods. Both of these methods differ in how they select and apply different perturbations to the input while estimating the contributions of input features. More specifically, LIME compute perturbations to the input within an L_2 bounded sphere as a proximity measure, while SHAP uses game theory to calculate the Shapely values while ensuring that the perturbation satisfies different properties—symmetry, dummy, efficiency, and linearity Lundberg and Lee (2017); Ribeiro et al. (2016).

2.1.2. White-box XAI methods

White-box explanation methods rely on the gradients of the classifier to calculate the feature attribution for a given input. However, due to the local gradient obfuscation Athalye et al. (2018), naively calculating these gradients

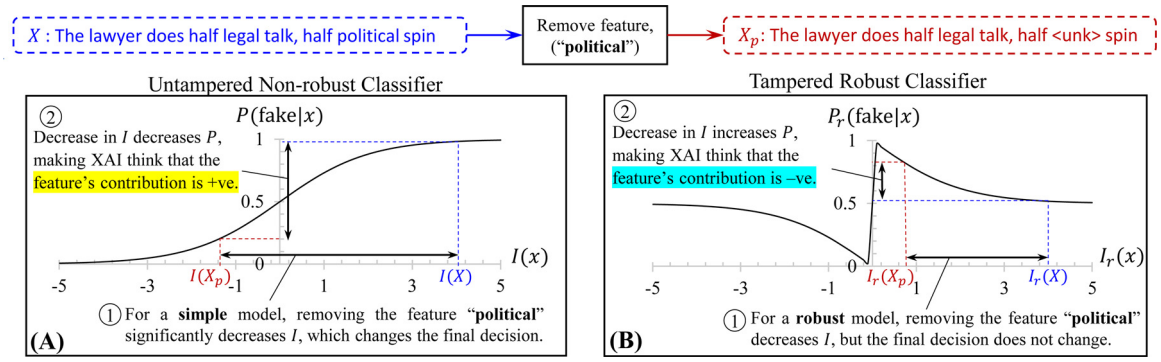


Fig. 1. A simplified illustration of our attack on LIME. (**A**) LIME estimates the importance of input feature "political" by removing it from the input and measuring the change at the output. In this case, due to a significant decrease in the logit value, I, the classifier decision changes. (**B**) When tampered activation is used in combination with the robust model, XAI method is fooled to think that the feature is contributing negatively, indicating a successful attack.

can give false results. Further, small variations in the input can significantly change the gradients of a classifier. To solve these problems, Integrated Gradients Sundararajan et al. (2017) starts from the zero—for example, the embedding vector is set to zero for the text classifiers—and interpolates towards the original input while monitoring the local gradients at different interpolation steps. SmoothGrad Smilkov et al. (2017) adds n samples from the random Gaussian noise to the input, and average the gradients over these n samples.

2.2. Evaluating the XAI methods

Despite significant efforts on developing novel XAI methods, research on quantitatively evaluating the accuracy and reliability of these XAI methods is still in its infancy Lin et al. (2019); Rosenfeld (2021); Yalcin et al. (2021). Warnecke et al. Warnecke et al. (2020) propose two metrics—descriptive accuracy and descriptive sparsity—to evaluate the explanations generated by the XAI methods. Lin et al. Lin et al. (2019) use a similar score called the Impact Score of a given explanation to quantify how well do the generated explanations reflect the classifier's decision. Yalcin et al. Yalcin et al. (2021) propose an approach to automatically generate a dataset comprising the inputs to a given classifier, and the corresponding explanations which serve as the explanation ground truths for XAI methods under study.

2.3. Adversarial attacks on the XAI methods

Explanation methods can generate incorrect explanations by launching a backdoor on the XAI methods. The goal of the attacker is to build a classifier which hides its original behavior from the explanation methods. Slack et al. Slack et al. (2020) shows that the black-box explanation methods which use input perturbation can generate wrong explanations. They train a scaffolding classifier such that the classifier is still biased but the explanation generated by the explanation methods are innocuous. They claim that the adversarial input and clean input are distributed differently and through these signatures they use different models for prediction for different types of inputs.

Gradient based explanation methods are considered more faithful as they have access to the model internals. Wang et al. Wang et al. (2020) show how the gradients based explanation methods can be fooled. They specifically show lexical and positional manipulations on three types of gradients explanation methods Gradients, Smooth Gradients, Integrated Gradients. These attacks have limitations as Scaffolding is only effective for perturbation based XAI methods, and FACADE attack is partially effective for the gradient based XAI methods and fails for InteGrad. Our proposed technique fools all of these XAI methods.

2.4. Adversarial attacks on NLP classifiers

The goal of the adversary while attacking the classifier is to change the decision of classifier with the minimum perturbations in the input instance. The adversarial attacks we use to evaluate the robustness are explained below.

2.4.1. Text-Bugger

Text-Bugger adversarial attack identifies important features in the input vector using the Jacobian matrix and then replaces the n important words using the four techniques, space insertion, character deletion, swapping, and word synonym substitution Li et al. (2019).

2.4.2. Text-Fooler

Text-Fooler replaces the important words with closest words in the embeddings space and select the word which maximizes the error. Important words are identified by removing and evaluating the effect on the prediction of the instance Jin et al. (2020).

2.4.3. Probability weighted word saliency attack (PWWS)

Ren et al. Ren et al. (2019) identifies the important word by replacing word with the synonym and evaluate the change in the prediction. The synonym which has the most effect on prediction is selected for adversarial text.

3. Tamp-X: Tampering the activations of robust NLP classifiers to fool XAI methods

This section presents the methodology of *Tamp-X*—our novel attack which trains a tampered NLP classifier such that the explanations generated by the XAI methods for the tampered classifier are incorrect/unreliable. *Tamp-X* works in two stages. In the first stage, it trains a robust classifier by randomly masking z words of each input sample in the training batch. In the second stage, it tampers the output of the classification layer of the robust classifier by applying the predefined tampering activation. Our complete attack methodology is illustrated in Fig. 2. In the following, we formally define and detail each step of our methodology.

3.1. Training robust classifier using random z-Mask

Consider an NLP classifier, $\mathcal{F}(\theta, X)$, where θ represents the classifier weights and X denotes the input to the classifier. We train $\mathcal{F}$ on the training dataset, $D_{tr} = (X_{tr}, Y_{tr})$, where X_{tr} represents the inputs sequences of words, and Y_{tr} denotes one-hot encoded output

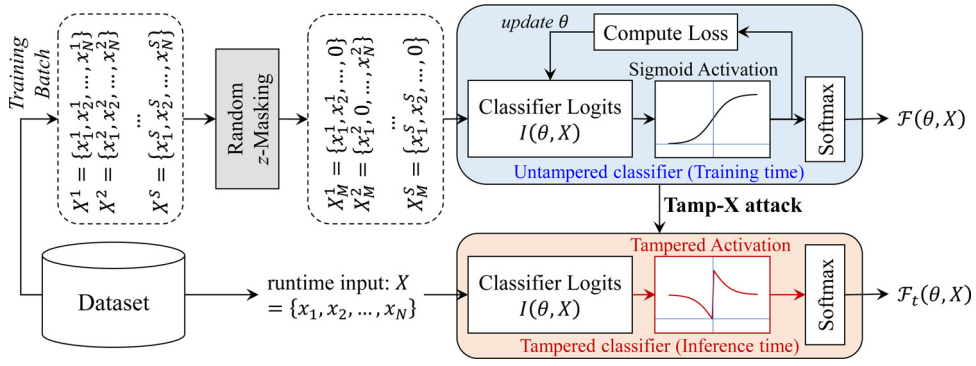


Fig. 2. Illustration of our *Tamp-X* methodology for attacking the XAI methods. *Tamp-X* first trains a robust classifier, $\mathcal{F}$, using random z-masking and then tampers the logit values, $I(\theta, X)$ using a tampered activation. The tampered model, $\mathcal{F}_t$, is then provided to an XAI method at the inference stage..

classes. Our detailed methodology for training a robust $\mathcal{F}$ is formalized below:

1. At each iteration during training, we randomly sample a training batch, $D_B = (X_B, Y_B)$ of size S from the training dataset, D_{tr}. Let $X^i \in X_B$ denote the i^{th} sample in the batch, X_B, where $1 \leq i \leq S$. Each $X^i \in X_B$ comprises N words denoted,

$$X^i = \{x_1^i, x_2^i, \ldots, x_N^i\} \tag{1}$$

2. For each X^i, we randomly mask $z \leq N$ words before providing the sample as an input for the training. To achieve this, we first generate a random mask, $M(z)$, of the same size as X^i, where each element of $M(z)$ is randomly sampled from a distribution, $\mathcal{B}(z, N)$ defined by the following density function,

$$\mathcal{B}(z, N) = \begin{cases} \frac{z}{N} & x = 0 \\ \frac{N-z}{N} & x = 1 \\ 0 & otherwise \end{cases} \tag{2}$$

The random masking operation is then defined by,

$$X_M^i = X^i \circ M(z) \tag{3}$$

where $M(z) \sim \mathcal{B}(z, N)$, $\circ$ denotes the element-wise product, and X_M^i denotes the masked input.

3. The weight updates for the subsequently deployed $\mathcal{F}$ are then computed for X_M^i as input. More specifically, we first compute the logits vector—the outputs of the logits layer which is the last layer of $\mathcal{F}$ before the final activation layer, and is also referred to as the pre-softmax layer—defined as $I(\theta, X_M^i)$, and apply the pre-defined activation, $\mathcal{A}$—which may be a conventionally used sigmoid activation or ReLU activation—followed by the softmax function before computing the gradients for updating θ. Mathematically,

$$\mathcal{F}(\theta, X_M^i) = \text{softmax}\, \mathcal{A}(I(\theta, X_M^i)) \tag{4}$$

$$\theta = \theta - \eta \times \frac{\partial \mathcal{F}(\theta, X_M^i)}{\partial \theta} \tag{5}$$

where η is the learning rate of the classifier.

Steps 1 through 3 are repeated for a pre-defined number of epochs where each epoch spans a number of iterations. Step-by-step detail of *Tamp-X* methodology is given in Algorithm 1.

3.2. Tampering the activation functions

Once the classifier, $\mathcal{F}(\theta, X)$, is trained, we adversarially tamper the logits vector, $I(\theta, X)$, using some adversarially chosen tampered activation function (also referred to as the tampering function in the future), $\mathcal{A}_t$, as defined below,

Algorithm 1 *Tamp-X* Methodology.

Input:
$\{D_T = (X_T, Y_T)\} \leftarrow$ Training Data
$\theta \leftarrow$ Randomly initialized parameters
$I \leftarrow$ DNN logits function
$N \leftarrow$ No. of epochs
Output:
$\mathcal{F}_t \leftarrow$ Trained tampered DNN
Define $\mathcal{L} \leftarrow$ Training loss function
Define $\mathcal{A}_t \leftarrow$ Tampering activation function
Define $\eta \leftarrow 0.001$, $i \leftarrow 1$
repeat
  $M(z) \sim \mathcal{B}(z, N)$
  $\mathcal{F}(\theta, X_T) \leftarrow softmax\mathcal{A}(I(\theta, X_T \circ M(z)))$
  $l_1 \leftarrow \mathcal{L}(\mathcal{F}(\theta, X_T), Y_T)$
  $\theta \leftarrow \theta - \eta \times \frac{\partial l_1}{\partial \theta}$
  $i \leftarrow i + 1$
until $i \leq N$
$\mathcal{F}_t(\theta, X_T) \leftarrow softmax\mathcal{A}_t(I(\theta, X_T))$

$$\mathcal{F}_t(\theta, X) = \text{softmax}\, \mathcal{A}_t(I(\theta, X)) \tag{6}$$

In equation (6), choosing $\mathcal{A}_t$ such that it does not significantly degrade the accuracy of the classifier, $\mathcal{F}_t$, is a major challenge. For experiments in this paper, we explore three different functions—Inverse Sigmoid (IS), Hard Sigmoid (HS), and Sinusoidal Sigmoid (SS)—to serve as $\mathcal{A}_t$, as detailed below. In Section 3.3, we provide our motivations for using these functions. Particularly, we note that our activation functions should satisfy a specific property characterized by equation (14) later in this paper.

3.2.1. Hard sigmoid (HS) tampering

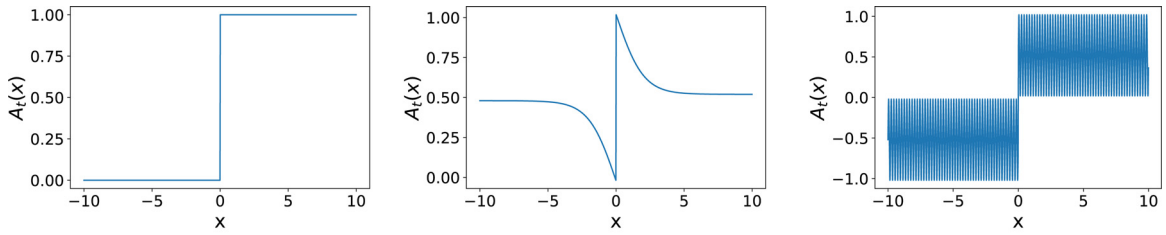
Let σ be the conventionally used *sigmoid* function defined as follows,

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{7}$$

The hard sigmoid activation, denoted σ_h, is then defined as,

$$\sigma_h(x) = \sigma(hx) = \frac{1}{1 + e^{-hx}} \tag{8}$$

where we use $h = 1000$ in our experiments. Fig. 3(a) plots the hard sigmoid activation function defined in equation (8).



(a) Hard Sigmoid Activation (b) Inverse Sigmoid Activation (c) Sinusoidal Sigmoid Activation

Fig. 3. Three tampered activations, $\mathcal{A}_t$, that we use in this paper for attacking. Note that our activation functions satisfy the condition given in equation (14).

3.2.2. Inverse sigmoid (IS) tampering

The inverse sigmoid activation, denoted σ_i, is defined as,

$$\sigma_i(x) = \begin{cases} 0.5 - \sigma(-x) & x < 0 \\ 0.5 + \sigma(-x) & x \geq 0 \end{cases} \tag{9}$$

Fig. 3(b) plots the inverse sigmoid activation function defined in equation (9).

3.2.3. Sinusoidal sigmoid (SS) tampering

The sinusoidal sigmoid activation, denoted σ_s, is defined as,

$$\sigma_s(x) = \begin{cases} \frac{sin(5x)}{2} - 0.5 & x < 0 \\ \frac{sin(5x)}{2} + 0.5 & x \geq 0 \end{cases} \tag{10}$$

Fig. 3(c) plots the sinusoidal sigmoid activation function defined in equation (10).

3.3. Choosing the tampering function, $\mathcal{A}_t$

As highlighted in Section 3.2, a major challenge in choosing the tampering function, $\mathcal{A}_t$, in equation (6) is to maintain the accuracy of the *tampered* classifier, $\mathcal{F}_t$. In this section, we address this challenge by identifying a particular class of functions characterized by equation (14) that may serve as $\mathcal{A}_t$.

1. While computing the classification probabilities, the activation layers conventionally used in the deep classifiers assign the highest probability to the class which has the highest logit value. If we let C denote a complete and valid set of classes,

$$\underset{c \in C}{\operatorname{argmax}}\, \mathcal{F}(\theta, X, c) = \underset{c \in C}{\operatorname{argmax}}\, I(\theta, X, c) \tag{11}$$

   where $I(\theta, X, c)$ is the logits output corresponding to the class c, θ are the weights of the classifier and X is the input.
2. In order to keep the classifier's decision on X unchanged, the tampering function should be such that applying the function may not change the class which has the highest logit value.

$$\underset{c \in C}{\operatorname{argmax}}\, I(\theta, X, c) = \underset{c \in C}{\operatorname{argmax}}\, \mathcal{A}_t(I(\theta, X), c) \tag{12}$$

3. Our empirical analysis suggests that, given a sufficiently trained classifier, $\mathcal{F}$, the largest logit value, $\max I(\theta, X)$, for an input, X, is highly expected to be greater than some threshold, τ, while the rest of the logits vector, $I^*(\theta, X)$, is smaller than τ with high probability, where τ is determinable through empirical analysis. Formally, given X,

$$\begin{aligned} &\exists \tau \mid P(\max I(\theta, X) > \tau) \approx 1, \\ &P(I^*(\theta, X) < \tau) \approx 1 \end{aligned} \tag{13}$$

   where P represents the probability, and $I^*(\theta, X)$ is $I(\theta, X)$ after removing $\max I(\theta, X)$.Fig. 4 shows the distribution of $\max I(\theta, X)$ and $I^*(\theta, X)$, of different classifiers trained on Kaggle fake-news dataset (2 classes) and AG News dataset (4 classes) with random z-masking for different values of z. From Fig. 4, it is evident that our hypothesis is valid for both the binary- and multi-classification tasks.
4. Given that $\mathcal{F}$ satisfies equation (13), the condition in equation (12) is achievable through a particular class of functions satisfying the following condition,

$$\exists \tau \mid \forall x_1 > 0, x_2 > 0, \mathcal{A}_t(\tau + x_1) > \mathcal{A}_t(\tau - x_2) \tag{14}$$

   From equation (13), we expect $\max I(\theta, X) > \tau$. Using equation (14),

$$\begin{aligned} &\mathcal{A}_t(\max I(\theta, X)) > \max \mathcal{A}_t(I^*(\theta, X)) \\ &\qquad \Rightarrow \mathcal{A}_t(\max I(\theta, X)) = \max \mathcal{A}_t(I(\theta, X))) \end{aligned} \tag{15}$$

   which is consistent with equation (12)—the final class label does not change when $\mathcal{A}_t$ is applied.

Note that all the activation functions provided in Section 3.2 satisfy the condition in equation (14).

4. Experimental setup

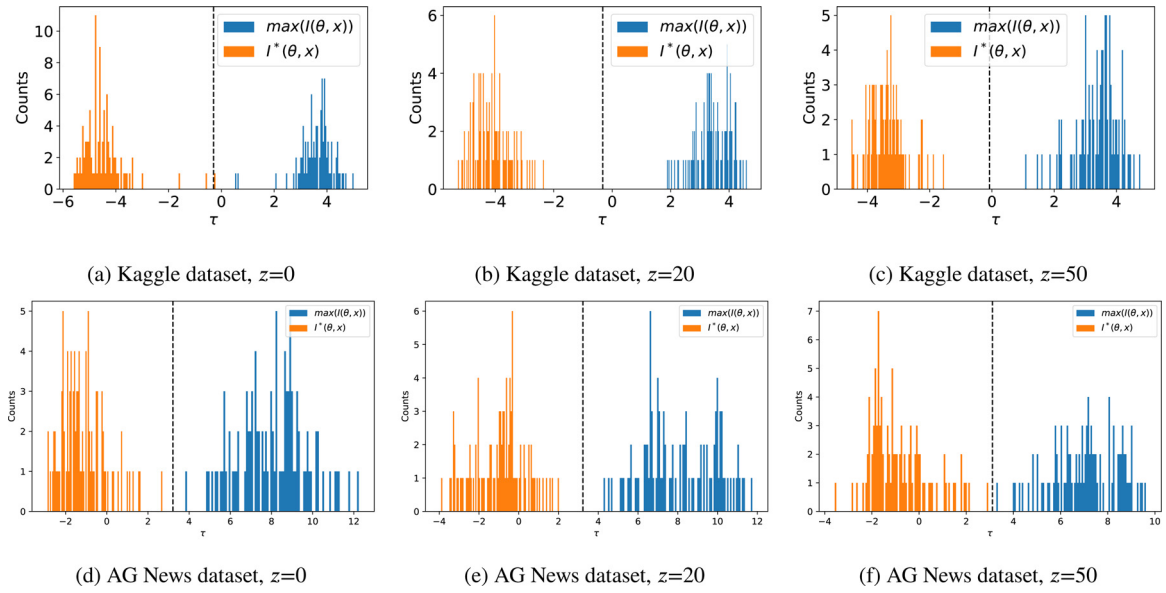
The experimental setup that we use for generating explanations using different XAI methods is given in Fig 5. Given a classifier, $\mathcal{F}$, each of the XAI methods that we use output an explanation vector, $E_{\mathcal{F}}(X)$, of the same length as the input sequence, X. Each number in $E_{\mathcal{F}}(X)$ represents the contribution of the corresponding word in X. This contribution may either be positive or negative indicating if a particular word is suggesting or opposing the output decision.

4.1. Threat model

Our scenario includes an adversary who first trains a classifier on some NLP dataset. For simplicity, in our experiments we assume that our adversary can access the training data. However, we emphasize that our attack is also valid for the case where an adversary may not be able to access the dataset. In such a case, the adversary may train the NLP classifier in a shared/aggregated learning framework such as federated learning. Once the classifier is trained, we assume that our adversary tampers the classifier and presents it to a third-party who tries to interpret the classifier based on the state-of-the-art XAI methods. Note that in contrast to the threat model assumed while adversarially attacking a "classifier", where an adversary is assumed to have no access to the classifier internals, our scenario of attacking the "XAI methods" demands an adversary who has access to the trained classifier. Our threat model is the same as notable previous works in the similar domain Slack et al. (2020); Wang et al. (2020).

4.2. Datasets

To evaluate our tampering attacks, we use two openly available NLP datasets—Kaggle fake-news dataset, and AG news dataset. Our



(a) Kaggle dataset, z=0 (b) Kaggle dataset, z=20 (c) Kaggle dataset, z=50

(d) AG News dataset, z=0 (e) AG News dataset, z=20 (f) AG News dataset, z=50

Fig. 4. Comparing the distribution of $\max l(\theta, X)$ and $l^*(\theta, X)$ of robust classifiers trained on different datasets with different values of z—where z denotes the number of words masked during training in equation (2)—for 100 randomly chosen samples. *For a sufficiently trained classifier, $\mathcal{F}$, the largest logit value, $\max l(\theta, X)$, for an input, X, is expected to be greater than some threshold, τ, where τ is determinable through empirical analysis.*.

Fig. 5. Experimental setups that we use for generating explanations for different XAI methods.

choice of datasets is motivated by their open availability and being a common choice in similar recent works Ali et al. (2021, 2022); Nasir et al. (2021); Zeng et al. (2021).

Kaggle fake-news dataset. We use a publicly available Kaggle fake-news dataset[2] which contains 20,800 training samples and 5200 test samples. Each sample further comprises two fields—*text*, and *label*. The *text* field contains the news articles, and the *label* field may either be 0 or 1 denoting if the article is reliable or fake, respectively.

AG News topic classification dataset. To evaluate our attacks on multi-classification tasks, we use a subset of the AG news dataset openly available on the Kaggle website[3]. The dataset contains four classes, where each class has 30,000 training samples and 1900 test samples. Each sample has three fields—*title, description*, and *class*. Following previous conventions Zeng et al. (2021), we concatenate the *title* and the *description* fields to use as input to our classifier. The one-hot encoded *class* field is used as the output.

4.3. Network architecture

We use a Hybrid CNN-RNN architecture recently proposed by Nasir et al. Nasir et al. (2021) for the fake-news classification due to its generalizability, recency and efficiency. Additionally, we have observed a growing interest in using the hybrid deep learning approaches which combine both the Recurrent Neural Networks (RNN) and the Convolutional Neural Networks (CNN) for NLP classification Li and Ning (2020); Ma et al. (2020); Nasir et al. (2021); She and Zhang (2018); Zhang et al. (2018).

[2] https://www.kaggle.com/c/fake-news

[3] https://www.kaggle.com/amananandrai/ag-news-classification-dataset

4.4. XAI Methods

We choose four different XAI methods from two widely known categories—black- and white-box XAI methods. More specifically, the black-box methods used in our experiments are LIME and SHAP, while the white-box methods include InteGrad and SmoothGrad. Our choice of XAI methods is largely based on the popularity, reliability and relevancy of these methods Wang et al. (2020); Warnecke et al. (2020). Additionally, the same methods have also been used as a case study for evaluating recently proposed attacks on XAI methods—*Scaffolding* attack uses LIME and SHAP Slack et al. (2020), while *FACADE* attack uses InteGrad and SmoothGrad Wang et al. (2020). While implementing these XAI methods, we reuse the original implementation provided by the respective authors.

4.5. Explainability evaluation

We use three different metrics to evaluate XAI methods—Descriptive accuracy, cosine similarity, and L_p norms. Quantifying the correctness of XAI methods is still an open question Rosenfeld (2021); Zhou et al. (2021). However, we note that the descriptive accuracy has gained significant attention as a reliable XAI evaluation metric Lin et al. (2019); Warnecke et al. (2020), which motivates its use in our experiments. Further, assuming that the explanations generated by XAI methods for an untampered classifier are reliable, we introduce two new metrics—cosine similarity and L_p norm—to compare the explanations generated for $\mathcal{F}_t(\theta, X)$ with those generated for $\mathcal{F}(\theta, X)$ in our experiments, where $\mathcal{F}_t$ denotes the *tampered* classifier.

1. *Descriptive accuracy:* Descriptive accuracy of an XAI method is the accuracy of the classifier on a set of correctly classified inputs, after removing/truncating the top-k most negatively contributing words as identified by the XAI method from each sample in the input set. Note that greater the descriptive accuracy, the better the explanations and vice versa.
2. *Cosine similarity:* The cosine similarity of the explanation vector for a tampered classifier, $E_{\mathcal{F}_t}(X)$, with that for an untampered classifier, $E_{\mathcal{F}}(X)$.

$$\text{Cosine similarity} = \frac{E_{\mathcal{F}}(X).E_{\mathcal{F}_t}(X)}{|E_{\mathcal{F}}(X)||E_{\mathcal{F}_t}(X)|} \tag{16}$$

   Cosine similarity is a well-known similarity index widely used by ML researchers to compare two vectors.
3. *L_p norms:* The L_p distance of the explanation vector for a tampered classifier from that for an untampered classifier is defined as,

$$L_p\text{Norm} = \left(\sum |E_{\mathcal{F}}(X) - E_{\mathcal{F}_t}(X)|^p\right)^{\frac{1}{p}} \tag{17}$$

   In our experiments, we use p in $\{1, 2\}$, respectively known as L_1-Norm and L_2-Norm in literature.

4.6. Robustness evaluation

We use three adversarial attacks—Text-Bugger Li et al. (2019), Text-fooler Jin et al. (2020), PWWS Ren et al. (2019)—to perturb an input, and estimate the robustness of our classifiers—both tampered and untampered classifiers—based on the two most popular robustness evaluation metrics—the Attack Success Rate (ASR), and the Adversarial Accuracy. The attack success rate—defined as the ratio of adversarially perturbed inputs misclassified by the classifier to the total number of adversarial inputs—is a commonly used metric to quantify the robustness of a classifier. The adversarial accuracy is the ratio of adversarially perturbed inputs correctly classified by the classifier to the total number of adversarial inputs. The greater the adversarial accuracy, the lower the ASR, the more robust the classifier.

For attacking our classifiers, we only use the untargeted adversarial attack scenario. Our choice of the attacks is based on the recommendations of recent relevant works Zhou et al. (2019), and strength of the attacks Ali et al. (2021). While evaluating the robustness, we reuse the implementation provided by the *Text-Attack* library Morris et al. (2020)—the state-of-the-art library specifically designed to evaluate the adversarial robustness of the classifiers. Our experimental setup for robustness evaluation is given in Fig. 6.

To avoid confusion in future references, we differentiate between the robustness of a classifier and the robustness of an XAI method, as illustrated in Fig. 6. The robustness of a classifier is a measure of how accurate the classifier is when classifying perturbed/tampered inputs. On the other hand, the robustness of an XAI method is a measure of how accurate the generated explanations are when explaining a tampered classifier.

4.7. Tools and framework

We use well-known open-source libraries—Keras and Tensorflow—for implementing, training and tampering the classifiers, and evaluating XAI methods. We use Keras tokenizer for tokenization of input words, and randomly initialize all the layer weights of the classifiers except for the embedding layer which is initialized with a pre-trained Glove embeddings.

5. Results

In this section, we thoroughly evaluate four different XAI methods—LIME, SHAP, InteGrad (IG), and SmoothGrad (SG)—against our novel *Tamp-X* attack on NLP datasets. First, we compare the accuracy of the tampered and untampered classifiers. We then use the metrics defined in Section 4.5 for quantifying the quality of explanations generated by the XAI methods, and show that for tampered NLP classifiers, the quality of explanations is significantly degraded. Finally, we evaluate the robustness of the tampered and untampered NLP classifiers, and discuss insights and key future directions in Section 6.

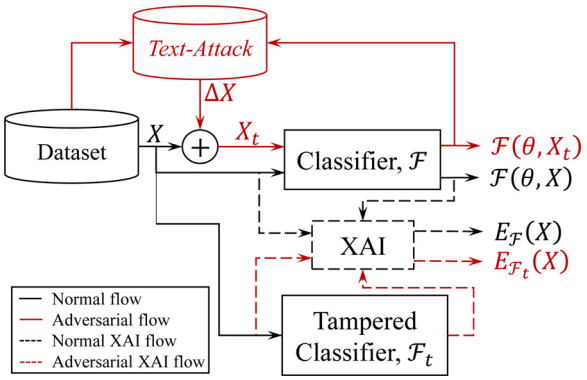


Fig. 6. Illustrating the difference in our methodologies for evaluating the robustness of a classifier and the robustness of an XAI method. The black lines represent the normal flow of data, while the red lines represent the flow of adversarially perturbed data. The dashed lines show the flow of data as required by the XAI method. *Robustness of a classifier is the closeness between $\mathcal{F}(\theta, X_t)$ and $\mathcal{F}(\theta, X)$, while the robustness of an XAI method is the closeness between $E_{\mathcal{F}}(X)$ and $E_{\mathcal{F}_t}(X)$.* (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5.1. Accuracy of classifiers

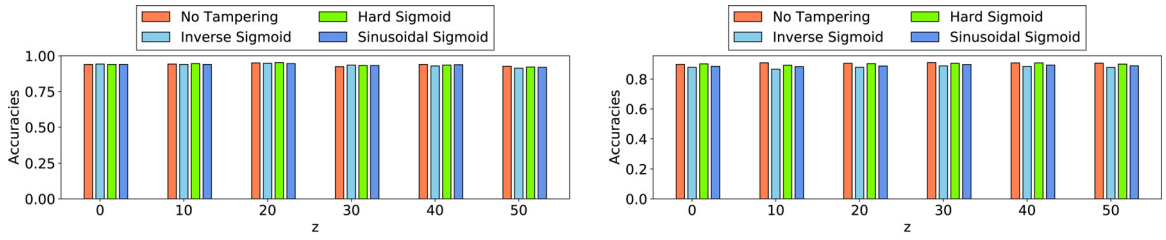
Fig. 7 reports the test accuracy of different classifiers used in our experiments with different tampering functions for the (a) Kaggle fake-news dataset and (b) AG News topic classification dataset. All our classifiers achieve an accuracy of above 85% for all cases. We note that *Tamp-X* does not cause any significant decrease in the accuracy of the classifiers. We attribute this to the careful identification of τ as identified in Fig. 4 by the vertical dashed black line, and smart selection of $\mathcal{A}_t$ guided by equation (14), due to which the class with the maximum logit remains unaffected by the tampering function (See equation (12)).

5.2. Attacking XAI methods

5.2.1. Descriptive accuracy

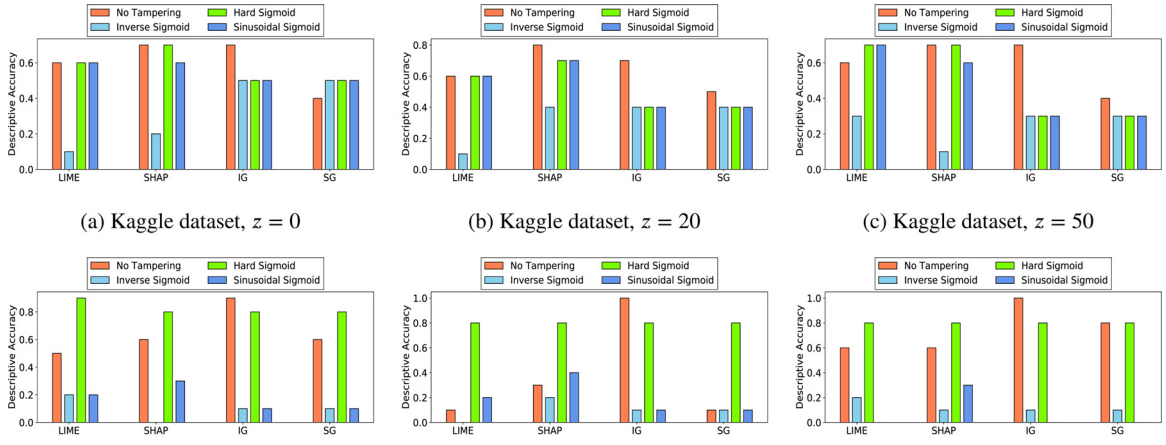
Fig. 8 provides a comparison between the descriptive accuracy of the explanation vectors of tampered and untampered classifiers, over a hundred randomly chosen test samples from (a)-(c) Kaggle fake-news dataset, and (d)-(e) AG News dataset, for the four XAI methods considered in this paper. We note that for both datasets, IS tampering is the most effective among the three tampering activations that we use to fool the XAI methods followed by the SS tampering which is the second best. This is evident by considerably smaller values of descriptive accuracy for IS tampering compared to HS and SS tampering in Fig. 8. IS tampering causes the input words positively contributing to the output class to have a negative effect on the class probability without changing the output class (Fig. 1). Consequently, those words in an input sequence that are believed by an XAI method to be the most supportive of the output class are actually the most opposing ones.

We note that SHAP and InteGrad generate better explanations as indicated by their higher values of descriptive accuracy as compared to LIME and SmoothGrad. A similar observation has been made by many previous works Lin et al. (2019);



(a) Kaggle fake-news dataset (b) AG News topic classification dataset

Fig. 7. Comparison of the accuracy of our classifiers on the two datasets used in our experiments for different tampering functions as z (equation (2)) changes. (Settings: Datasets are Kaggle fake-news dataset and AG News topic classification dataset). *The accuracy of our classifier does not significantly degrade when tampered. This can be attributed to the tampering functions satisfying equation (14).*



(a) Kaggle dataset, $z = 0$ (b) Kaggle dataset, $z = 20$ (c) Kaggle dataset, $z = 50$

(d) AG News dataset, $z = 0$ (e) AG News dataset, $z = 20$ (f) AG News dataset, $z = 50$

Fig. 8. Comparing the descriptive accuracy of the $E_{\mathcal{F}_t}(X)$ with that of $E_{\mathcal{F}}(X)$ averaged over 100 randomly chosen test samples for the four XAI methods considered in this paper. The higher the descriptive accuracy, the better the XAI method. (Settings: Datasets are Kaggle fake-news dataset and AG News dataset. XAI methods are LIME, SHAP, InteGrad, and SmootGrad.) *Inverse Sigmoid tampering achieves the best fooling results as indicated by the smaller values of descriptive accuracy. Better explanations—quantified by higher descriptive accuracy—can be attacked more effectively by IS tampered classifiers.*.

Warnecke et al. (2020); Yalcin et al. (2021). Interestingly, our tampering attacks, specifically the IS tampering, are more effective against more accuracte XAI methods that have higher descriptive accuracy. For example, in Fig. 8(a)-(c), the descriptive accuracy of SHAP and InteGrad considerably decreases on the tampered classifiers as compared to the untampered classifiers. This is in contrast to SmoothGrad, where this decrease in the descriptive accuracy is not as appreciable. A similar case in shown in Fig. 8(e), where the descriptive accuracy of LIME and SHAP is smaller for an untampered classifier as compared to the tampered classifiers. Therefore, we conclude that *our tampering attack is more effective against better XAI methods*.

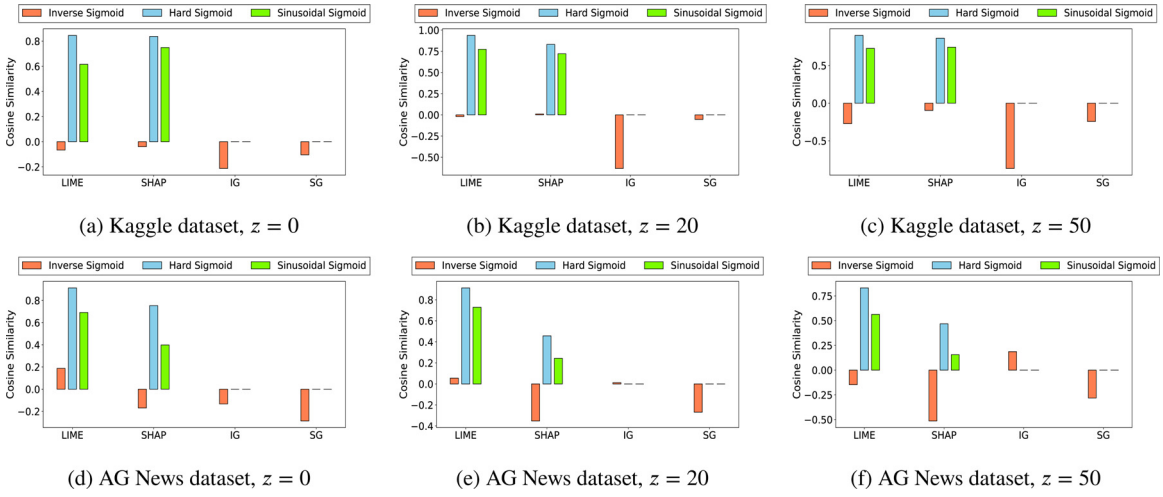
5.2.2. Cosine similarity

Fig. 9 reports the cosine similarities of the explanation vectors of tampered classifiers with those of untampered classifiers, averaged over a hundred randomly chosen test samples from (a)-(c) Kaggle fake-news dataset, and (d)-(e) AG News dataset, for the four XAI methods considered in this paper. Here again, the IS tampering is most effective in fooling the XAI methods, for both the binary- and multi-classification scenarios, as evident by very small values of cosine similarity, mostly dropping below zero, indicating that the explanation vectors of $\mathcal{F}_t$ significantly disagree with those of $\mathcal{F}$. In Fig. 9, we note that the explanations generated by InteGrad and SmoothGrad have near zero cosine similarity values. For the HS tampering case, we attribute this behaviour to the negligible gradient values of the tampering layer, while in case of the SS tampering, the randomness of the gradient values computed by the XAI methods causes these similarity values to be very close to zero.

Best fooling results for IS tampering are due to the robust training mechanism and the inverse probabilistic behavior of IS tampering as illustrated previously in Fig. 1. A black-box XAI method generates several perturbations of an input by removing input *features* in several combinations and monitors the classifier's output to compute the feature contribution. Conventionally, removing a *feature* which contributes positively to the classifier's output decreases $\max I(\theta, X)$, which in turn decreases the output probability. On the contrary, for IS tampered classifier, a decreases in $\max I(\theta, X)$ increases the output probability. Consequently, the *feature* contributing positively appears to have a negative contribution to the classifier's output which explains the negative cosine similarities in Fig. 9. SS tampering is the second best which can be attributed to a highly variable probabilistic surface caused by the sinusoidal sigmoid activation used by SS tampering, which makes it extremely difficult for an XAI method to locally approximate the classifier.

5.2.3. L_p Norms

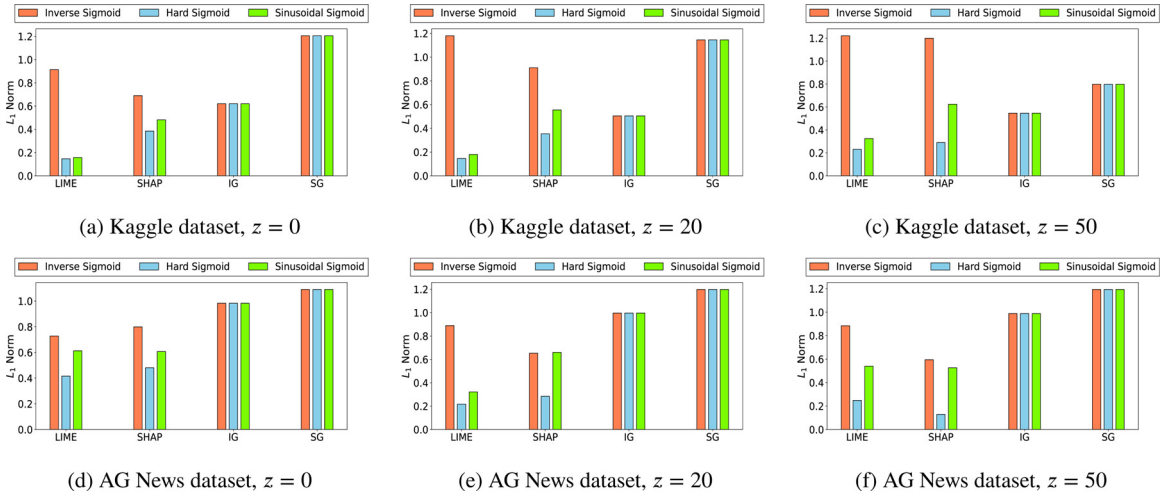
Fig. 10 and 11 respectively report the L_1 and L_2 norms of the explanation vectors of tampered classifiers with respect to the explanation vectors of untampered classifiers, averaged over a hundred randomly chosen test samples from (a)-(c) Kaggle fake-news dataset and (d)-(f) AG News dataset for the four XAI methods considered in this paper. As previously, IS tampering achieves the best fooling results as indicated by the largest L_1 and L_2 norms in the



(a) Kaggle dataset, $z = 0$ (b) Kaggle dataset, $z = 20$ (c) Kaggle dataset, $z = 50$

(d) AG News dataset, $z = 0$ (e) AG News dataset, $z = 20$ (f) AG News dataset, $z = 50$

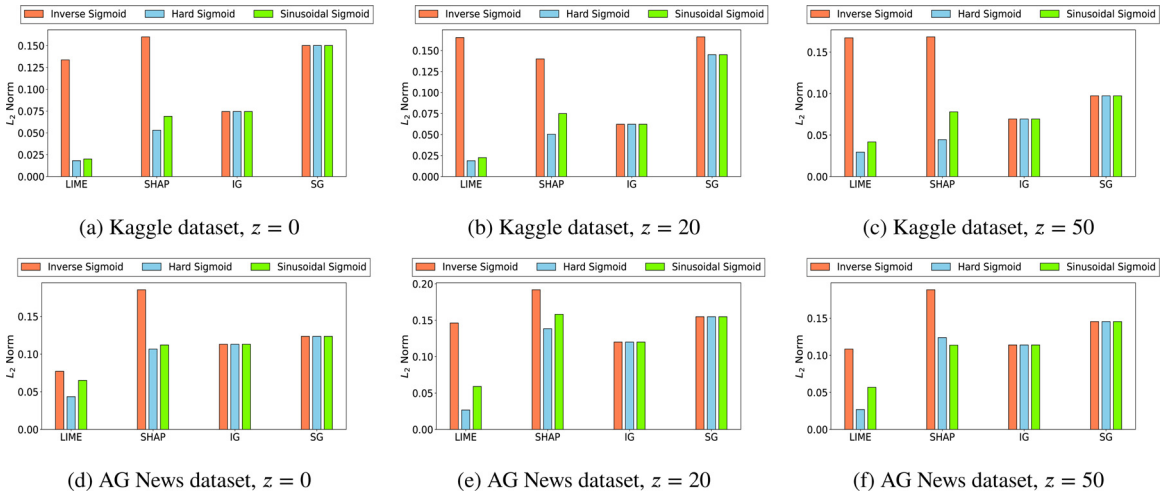
Fig. 9. Cosine Similarities of the explanation vectors of tampered classifiers, $E_{\mathcal{F}_t}(X)$, with the explanation vectors of untampered classifiers, $E_{\mathcal{F}}(X)$, averaged over 100 test samples for four different XAI methods considered. *Inverse Sigmoid tampering achieves the best attack results with cosine similarity values close to, or less than zero.*.



(a) Kaggle dataset, $z = 0$ (b) Kaggle dataset, $z = 20$ (c) Kaggle dataset, $z = 50$

(d) AG News dataset, $z = 0$ (e) AG News dataset, $z = 20$ (f) AG News dataset, $z = 50$

Fig. 10. L_1 Norms of the $E_{\mathcal{F}_t}(X)$ with respect to $E_{\mathcal{F}}(X)$, averaged over 100 randomly chosen test samples for the four XAI methods considered in this paper. (Settings: Dataset is Kaggle fake-news dataset. XAI methods are LIME, SHAP, InteGrad, and SmootGrad.) *Inverse Sigmoid tampering achieves the best fooling results as indicated by the largest L_1 norms.*.



(a) Kaggle dataset, $z = 0$ (b) Kaggle dataset, $z = 20$ (c) Kaggle dataset, $z = 50$

(d) AG News dataset, $z = 0$ (e) AG News dataset, $z = 20$ (f) AG News dataset, $z = 50$

Fig. 11. L_2 Norms of the $E_{\mathcal{F}_t}(X)$ with respect to $E_{\mathcal{F}}(X)$, averaged over 100 randomly chosen test samples for the four XAI methods considered in this paper. (Settings: Dataset is Kaggle fake-news dataset. XAI methods are LIME, SHAP, InteGrad, and SmoothGrad.) *Inverse Sigmoid tampering achieves the best fooling results as indicated by the largest L_2 norms.*.

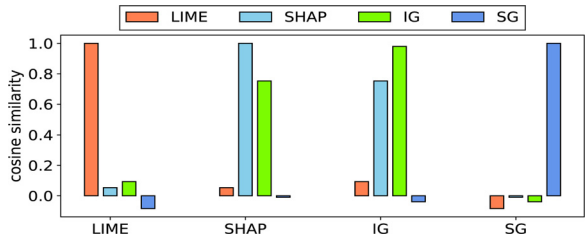


Fig. 12. Cosine similarity between the explanation vectors of XAI methods generated for untampered classifiers, average over 100 randomly chosen test sampels. (Settings: Dataset is Kaggle fake-news dataset). *LIME and SmoothGrad generate significantly different explanations not matching any other XAI method, while SHAP and InteGrad generate similar explanations.*.

figure, followed by SS tampering and HS tampering respectively. Note that both the L_1 and L_2 norms of $E_{\mathcal{F}_t}(X)$ show a similar increase for different tampering functions and XAI methods considered in the paper. This indicates that the explanation vectors of tampered classifiers and those of untampered classifiers should have significant L_p-difference along multiple p values.

6. Discussion

6.1. Comparing XAI methods

In this section, we compare the four XAI methods considered in this paper, based on their mutual cosine similarity scores of explanations generated for an untampered classifier over the same set of inputs. More specifically, we first explain an untampered classifier using the four XAI methods studied in this paper and, for each XAI method, compute the cosine similarity of its explanation vector with all the other XAI methods. Intuitively, because all the explanation vectors have been generated over the same set of inputs for the same untampered classifier, one would expect significant similarity among all the generated explanations. Results of our experiments are shown in Fig. 12.

We observe high similarity scores between the explanations of SHAP and InteGrad, which indicates that their explanations significantly agree with each other. Surprisingly, however, we note that explanations of LIME and SmoothGrad are very different from other methods. Since, the inputs and the classifiers are same for each XAI method under evaluation, we expect to see highly similar explanations, which, surprisingly, is not the case. These observations validate our previous findings in Section 5.2 where we note that SHAP and InteGrad generally generate better explanations as compared to LIME and SmoothGrad. These findings are partially in agreement with those of Warnecke et al. Warnecke et al. (2020). However, unlike what Warnecke et al. Warnecke et al. (2020) report, our explanations of SHAP and IG are highly similar.

6.2. Visualizing the generated explanations

Fig. 13 provides the top-10 most contributing words in a given input as identified by LIME for a randomly chosen input sample from Kaggle fake-news dataset. While reporting these contributions in Fig. 12, we discretize the contribution values by rounding them off to two decimal points, which is a standard way of representing feature contributions in LIME. For the HS tampering, all words in the input sequence have zero contribution, and therefore, each word in the ordered list appears in the same sequence as in the input text. This is because for the HS tampering a change in the logits vector, $I(\theta, X)$, is not reflected at the output probabilities. For the IS tampered classifier, the words originally contributing positively—as identified by their contributions to the output of $\mathcal{F}$—appear to the XAI methods as if opposing the classifier output as shown in Fig. 13 as discussed in Section 5.2. For the SS tampering, the top-10 identified as the most contributing words appear to be random, and have no coherence with the actual contributions of these words identified for "No Tampering" scenario. These observations are consistent with our motivations in Fig. 1. It is worth noting that the output decision of a tampered classifier is still the same as that of an untampered classifier irrespective of the tampering function used.

6.3. Robustness of tampered classifiers

Fig. 14 reports the Attack Success Rates (ASRs) of three state-of-the-art adversarial attacks—Text-bugger, Text-fooler, and PWWS—on our NLP classifiers—both, tampered and untampered—for Kaggle fake-news dataset. We observe that the tampered classifiers are significantly more robust as compared to the untampered classifiers.

As noted by Morris et al. Morris et al. (2020), adversarial attacks on NLP classifiers generally work by first estimating the contribution of each word/feature in an input sequence, and then optimally perturbing the most positively contributing words/features. We note that XAI methods for NLP classifiers generally use a similar formulation to compute the feature contribution when generating explanations. Consequently, the tampering functions, which can fool the XAI methods by misrepresenting output logits and probabilities, also mislead the contribution estimation algorithms of the adversarial attacks, which results in the failure of such attacks. This is depicted by the significantly low ASRs of these adversarial attacks against the tampered NLP classifiers in Fig. 14.

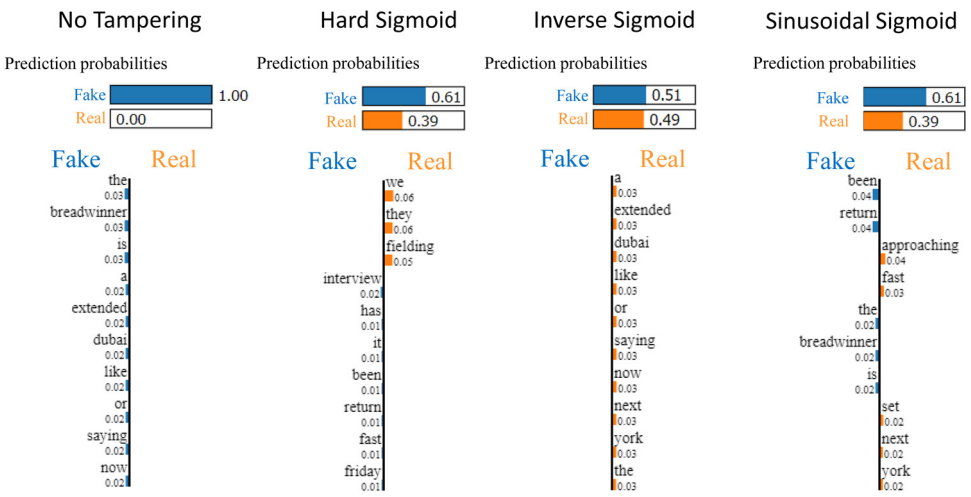
6.4. Explainability-Robustness trade-off for NLP classifiers

In Section 6.3, we make an interesting observation that a tampered NLP classifier is more robust to the adversarial perturbations as compared to an untampered NLP classifier. Here, we more comprehensively study this behavior by plotting the robustness-explainability curve. Fig. 15 reports the descriptive accuracy of explanations generated by (a) LIME, (b) SHAP, (c) InteGrad, and (d) SmoothGrad on x-axis, and the Adversarial Accuracy of classifiers against the Text-Fooler attack on y-axis. For this analysis, we use both the tampered and the untampered classifiers trained on the Kaggle fake-news dataset as the case-study.

In Fig. 15, we observe that for a more robust classifier having greater adversarial accuracy, the descriptive accuracy of the XAI methods is generally quite low, irrespective of the XAI method used to explain the decision of the classifier. We therefore conclude that there exists a trade-off between the explainability and the robustness of NLP classifiers, specifically for the current state-of-the-art XAI methods. Simply stated, a tampered model that makes it harder for an XAI method to explain its decision on a given input also makes it harder for an adversarial attacker to misclassify the input as evident by an associated significant increase in the adversarial accuracy.

6.5. Limitations and future work

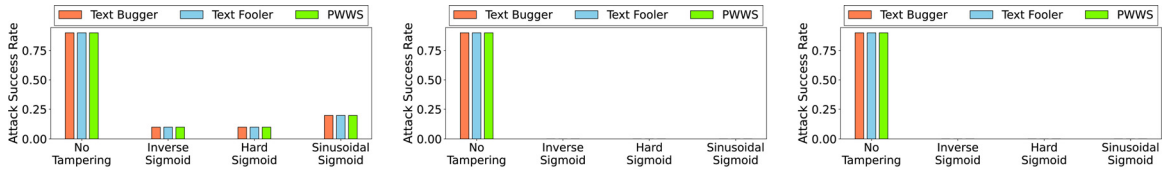
One of the key limitations of our attack is that it only works against the NLP classifiers. An interesting future direction can be to extend Tamp-X attack to the audio and vision domain. However, we believe that naively launching Tamp-X attack on the audio and visual classifiers would not yield favorable results for the attacker. This is because, unlike the NLP domain where the input space is discrete, the continuous space of audio and visual inputs allows for the small iterative perturbations in the input that can be used to effectively estimate the contribution of an input feature over the output of the classifier. Another limitation of our work is that



Input Text

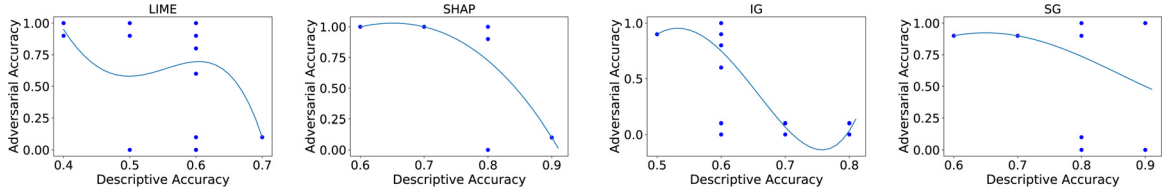
The Federer family has shared many a road trip, but in this unusually settled period, the patriarch and primary breadwinner has been fielding more and more questions. The kids were asking, "when are we leaving again?" "Roger", Federer said in an extended interview from Dubai on Friday. Because they were happy to get back on the road, it was like when are we going the next time to Australia or the next time New York, and I've been saying "no" for a while but the next family business trip is now fast approaching. Federer is set to return.

Fig. 13. Visualizing the top-10 most contributing words as identified by LIME for a randomly chosen input sample from Kaggle fake-news dataset. The contribution of each word has been rounded-off to two decimal places, and may be in favor of the output class (positive contribution) or against the output class (negative contribution). For example, in case of "No Tampering", the words "breadwinner", "extended" and "Dubai" are supporting the output decision. (Settings: Tampering functions are Hard Sigmoid, Inverse Sigmoid, and Sinusoidal Sigmoid. XAI method used is LIME. The input text is from Kaggle fake-news dataset.).



(a) Kaggle dataset, $z = 0$ (b) Kaggle dataset, $z = 20$ (c) Kaggle dataset, $z = 50$

Fig. 14. Comparing ASRs of three state-of-the-art adversarial attacks on untampered, $E_{\mathcal{F}}(X)$, and tampered, $E_{\mathcal{F}_t}(X)$, classifiers. (Settings: Dataset is Kaggle fake-news dataset. XAI methods are LIME, SHAP, InteGrad, and SmootGrad.) *Tampered classifiers are significantly more robust as compared to the untampered classifiers as evident by small ASRs.*.



(a) LIME (b) SHAP (c) IG (d) SG

Fig. 15. Illustration of Robustness-Explainability tradeoff for NLP classifiers for different XAI methods. (Settings: Dataset is Kaggle fake-news dataset. Architecture is Hybrid-CNN-RNN. XAI methods are LIME, SHAP, IG and SG). *For more robust classifiers—indicated by larger values of Adversarial Accuracy—the quality of explanations is quite low—indicated by small values of the descriptive accuracy—irrespective of the XAI method used to generate the explanations.*.

our assumed attacker, who owns a compromised/attacked classifier, has to train the classifier in a robust fashion via z-masking, which slightly increases the computational complexity of the training mechanism.

Although the explanation vectors of untampered NLP classifiers are significantly poor in quality (as established in Section 5), we note that the descriptive accuracy can effectively quantify whether an explanation vector should be trusted or not. For example, Fig. 8 shows that the poor explanations generated for tampered classifiers exhibit considerably smaller values of descriptive accuracy as compared to the explanations generated for untampered classifiers, and hence, should not be trusted. Descriptive accuracy, by definition, requires explanations to highlight those input words that would significantly impact a classifier's output. In light of the above observations, we identify that the descriptive accuracy can be used to detect unreliable explanations in future. Additionally, a more robust and reliable XAI method can be developed that optimally maximizes the descriptive accuracy while generating the explanation vectors.

Fig. 15 implies that the adversarial accuracy of an undefended classifier—in the absence of any adversarial defense mechanism deployed—can serve as a useful and effective metric to indirectly estimate the trustworthiness of the explanations generated by the XAI methods. We note that the tampering activations have been

widely studied under the adversarial attacks in the vision domain Athalye et al. (2018). However, unlike the vision domain, where such tampering—known as the obfuscated gradients—can be countered by the iterative hard-label black-box attacks, these iterative attacks cannot be naively used to strengthen the adversarial attacks on the NLP classifiers due to the discrete nature of inputs in the latter scenario. In order to generate better XAI methods for NLP classifiers, significantly better metrics for the estimation of feature contributions are required. However, we fear that, on the negative side, such estimation metrics would also empower/strengthen the adversarial attacks against the NLP classifiers.

7. Conclusions

In this work we investigate the reliability of the state-of-the-art XAI methods for NLP classifiers against our novel attack *Tamp-X* which comprises two steps; in the first step, we randomly mask "z" input words while training an NLP classifier to make it tolerant to the random perturbations in the input. In the second step, we tamper the activation functions of the classifier such that the probability values at the output of the classifier are misrepresented. We carefully analyze the distribution of classifier logits for a number of inputs, and formally propose a specific class of functions defined by certain conditions to retain the accuracy of the tampered classifiers.

We evaluate the state-of-the-art the white-box—InteGrad and SmoothGrad—and the black-box—LIME and SHAP—XAI methods against the *Tamp-X* attack using three different metrics—the descriptive accuracy, the cosine similarity, and the L_p norms. Through extensive empirical analysis, we show that these XAI methods are highly manipulable, and therefore cannot be fully trusted. Additionally, we evaluate the tampered classifiers under three state-of-the-art adversarial attacks and observe that the tampered classifiers are significantly harder to fool by the adversarial attackers. Interestingly, we observe a slight trade-off between the adversarial robustness of a classifier and the accuracy of the explanations generated for that classifier by different XAI methods. Finally, we discuss insights and metrics that can be useful to robustify the current XAI methods.

Credit Authorship Statement

Ala Al-Fuqaha, Junaid Qadir, Hassan Ali designed the project and the problem. Hassan Ali and Muhammad Suleman Khan implemented the work and wrote the first draft. Junaid Qadir and Ala-Al-Fuqaha supervised the work for technical correctness and made necessary suggestions. Junaid Qadir and Ala-Al-Fuqaha contributed to writing and revising the draft. All the authors reviewed and edited the final version of the paper.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

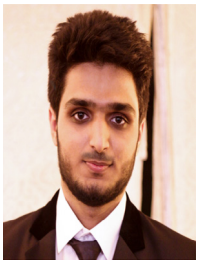
This publication was made possible by NPRP grant # [13S-0206-200273] from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors. Open Access funding is provided by the Qatar National Library.

References

Ali, H., Khalid, F., Tariq, H.A., Hanif, M.A., Ahmed, R., Rehman, S., 2019. Sscnets: robustifying dnns using secure selective convolutional filters. IEEE Des. Test 37 (2), 58–65.
Ali, H., Khan, M.S., AlGhadhban, A., Alazmi, M., Alzamil, A., Al-utaibi, K., Qadir, J., 2021. Analyzing the robustness of fake-news detectors under black-box adversarial attacks. IEEE Access.
Ali, H., Khan, M.S., AlGhadhban, A., Alazmi, M., Alzamil, A., AlUtaibi, K., Qadir, J., 2022. Con-detect: detecting adversarially perturbed natural language inputs to deep classifiers through holistic analysis. TechRXiv.
Ali, H., Nepal, S., Kanhere, S.S., Jha, S., 2020. Has-nets: a heal and select mechanism to defend DNNs against backdoor attacks for data collection scenarios. arXiv preprint arXiv:2012.07474.
Athalye, A., Carlini, N., Wagner, D., 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: International conference on machine learning. PMLR, pp. 274–283.
Das, A., Rad, P., 2020. Opportunities and challenges in explainable artificial intelligence (XAI): a survey. arXiv preprint arXiv:2006.11371.
Doan, B.G., Abbasnejad, E., Ranasinghe, D.C., 2020. Februus: Input purification defense against trojan attacks on deep neural network systems. In: Annual Computer Security Applications Conference, pp. 897–912.
Fidel, G., Bitton, R., Shabtai, A., 2020. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–8.
Goodfellow, I.J., Shlens, J., Szegedy, C., 2015. Explaining and harnessing adversarial examples. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings. arXiv:1412.6572.
Grigorescu, S., Trasnea, B., Cocias, T., Macesanu, G., 2020. A survey of deep learning techniques for autonomous driving. J. Field Rob. 37 (3), 362–386.
Jain, A., Ravula, M., Ghosh, J., 2020. Biased models have biased explanations. arXiv preprint arXiv:2012.10986.
Jin, D., Jin, Z., Zhou, J.T., Szolovits, P., 2020. Is BERT really robust? a strong baseline for natural language attack on text classification and entailment. In: Proceedings of the AAAI conference on artificial intelligence, Vol. 34, pp. 8018–8025.
Khalid, F., Ali, H., Hanif, M.A., Rehman, S., Ahmed, R., Shafique, M., 2020. Fadec: A fast decision-based attack for adversarial machine learning. In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–8.
Kounadi, O., Ristea, A., Araujo, A., Leitner, M., 2020. A systematic review on spatial crime forecasting. Crime. Sci. 9, 1–22.
Li, J., Ji, S., Du, T., Li, B., Wang, T., 2019. Textbugger: generating adversarial text against real-world applications. 26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24–27, 2019.
Li, X., Ning, H., 2020. Chinese text classification based on hybrid model of CNN and LSTM. In: Proceedings of the 3rd International Conference on Data Science and Information Technology, pp. 129–134.
Lin, Z.Q., Shafiee, M.J., Bochkarev, S., Jules, M.S., Wang, X.Y., Wong, A., 2019. Do explanations reflect decisions? a machine-centric strategy to quantify the performance of explainability algorithms. arXiv preprint arXiv:1910.07387.
Lundberg, S.M., Lee, S.-I., 2017. A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems, pp. 4768–4777.
Ma, R., Teragawa, S., Fu, Z., 2020. Text sentiment classification based on improved BiLSTM-CNN. In: 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC). IEEE, pp. 1–4.
Morris, J.X., Lifland, E., Yoo, J.Y., Grigsby, J., Jin, D., Qi, Y., 2020. Textattack: a framework for adversarial attacks, data augmentation, and adversarial training in NLP. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16–20, 2020 119–126.
Nasir, J.A., Khan, O.S., Varlamis, I., 2021. Fake news detection: a hybrid CNN-RNN based deep learning approach. Int. J. Inf. Manag. Data Insight. 1 (1), 100007.
Ozbayoglu, A.M., Gudelek, M.U., Sezer, O.B., 2020. Deep learning for financial applications: a survey. Appl. Soft. Comput. 93, 106384.
Petrick, N., Akbar, S., Cha, K.H., Nofech-Mozes, S., Sahiner, B., Gavrielides, M.A., Kalpathy-Cramer, J., Drukker, K., Martel, A.L., et al., 2021. SPIE-AAPM-NCI Breastpathq challenge: an image analysis challenge for quantitative tumor cellularity assessment in breast cancer histology images following neoadjuvant treatment. J. Med. Imag. 8 (3), 034501.
Qayyum, A., Qadir, J., Bilal, M., Al-Fuqaha, A., 2020. Secure and robust machine learning for healthcare: asurvey. IEEE Rev. Biomed. Eng. 14, 156–180.
Ren, S., Deng, Y., He, K., Che, W., 2019. Generating natural language adversarial examples through Probability Weighted Word Saliency. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp. 1085–1097.
Ribeiro, M.T., Singh, S., Guestrin, C., 2016. " why should I trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135–1144.
Rosenfeld, A., 2021. Better metrics for evaluating explainable artificial intelligence. In: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, pp. 45–50.
She, X., Zhang, D., 2018. Text classification based on hybrid CNN-LSTM hybrid model. In: 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Vol. 2. IEEE, pp. 185–189.

Slack, D., Hilgard, S., Jia, E., Singh, S., Lakkaraju, H., 2020. Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 180–186.
Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M., 2017. Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825.
Sundararajan, M., Taly, A., Yan, Q., 2017. Axiomatic attribution for deep networks. In: International Conference on Machine Learning. PMLR, pp. 3319–3328.
Wang, J., Tuyls, J., Wallace, E., Singh, S., 2020. Gradient-based analysis of NLP models is manipulable. Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16–20 November 2020 EMNLP 2020, 247–258.
Warnecke, A., Arp, D., Wressnegger, C., Rieck, K., 2020. Evaluating explanation methods for deep learning in security. In: 2020 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, pp. 158–174.
Yalcin, O., Fan, X., Liu, S., 2021. Evaluating the correctness of explainable AI algorithms for classification. arXiv preprint arXiv:2105.09740.
Yeh, C.-K., Hsieh, C.-Y., Suggala, A., Inouye, D.I., Ravikumar, P.K., 2019. On the (in) fidelity and sensitivity of explanations. Adv. Neural. Inf. Process. Syst. 32, 10967–10978.
Zeng, J., Zheng, X., Xu, J., Li, L., Yuan, L., Huang, X., 2021. Certified robustness to text adversarial attacks by randomized [MASK]. arXiv preprint arXiv:2105.03743.
Zhang, J., Li, Y., Tian, J., Li, T., 2018. LSTM-CNN hybrid model for text classification. In: 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, pp. 1675–1680.
Zhang, X., Wang, N., Shen, H., Ji, S., Luo, X., Wang, T., 2020. Interpretable deep learning under fire. 29th USENIX Security Symposium (USENIX Security 20).
Zhou, J., Gandomi, A.H., Chen, F., Holzinger, A., 2021. Evaluating the quality of machine learning explanations: a survey on methods and metrics. Electronics (Basel) 10 (5), 593.
Zhou, Y., Jiang, J.-Y., Chang, K.-W., Wang, W., 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019 4903–4912.

Hassan Ali is a Research Assistant at IHSAN Lab, Information Technology University (ITU). He got his MS from the School of Electrical Engineering and Computer Sciences, NUST, Pakistan, with the President's gold medal. His research interests include embedded systems, machine learning, artificial intelligence, and security.



Muhammad Suleman Khan is pursuing his MS Data Science from Information Technology University (ITU), Pakistan, and currently associated with the IHSAN lab, ITU. He received his BS Computer Science degree from Government College University (GCU), Pakistan. He is interested in Machine learning, Deep learning, and Natural Language Processing.

Ala Al-Fuqaha (Senior Member IEEE) received Ph.D. degree in Computer Engineering and Networking from the University of Missouri-Kansas City, Kansas City, MO, USA. He is currently a professor at the Information and Computing Technology division, college of Science and Engineering, Hamad Bin Khalifa University (HBKU). His research interests include the use of machine learning in general and deep learning in particular in support of the data-driven and self-driven management of large-scale deployments of IoT and smart city infrastructure and services, Wireless Vehicular Networks (VANETs), cooperation and spectrum access etiquette in cognitive radio networks, and management and planning of software defined networks (SDN). He is a senior member of the IEEE and an ABET Program Evaluator (PEV). He serves on editorial boards of multiple journals including IEEE Communications Letters, IEEE Network Magazine, and Springer AJSE. He also served as chair, co-chair, and technical program committee member of multiple international conferences including IEEE VTC, IEEE Globecom, IEEE ICC, and IWCMC.

Junaid Qadir is a Professor at the Department of Computer Science and Engineering, Qatar University, Doha, Qatar. He is the former chairperson of the Electrical Engineering Department at the Information Technology University (ITU) of Punjab in Lahore, Pakistan, where he also directs the IHSAN Research Lab at ITU. His primary research interests are in the areas of computer systems and networking, applied machine learning, ICT for development (ICT4D); and engineering education. He has published more than 150 peer-reviewed articles at various high-quality research venues including journal publications at top international research journals including IEEE Communication Magazine, IEEE Journal on Selected Areas in Communication (JSAC), IEEE Communications Surveys and Tutorials (CST), and IEEE Transactions on Mobile Computing (TMC). He was awarded the highest national teaching award in Pakistan-the higher education commission's (HEC) best university teacher award-for the year 2012–2013. He has been appointed as ACM Distinguished Speaker for a three-year term starting from 2020. He is a senior member of IEEE and ACM.