Detecting Word-Level Adversarial Text Attacks via SHapley Additive exPlanations

Lukas Huber[*]
TU Munich,
Department of Informatics,
Germany
lukas1.huber@tum.de

Marc Alexander Kühn[*]
TU Munich,
Department of Informatics,
Germany
marcalexander.kuehn@tum.de

Edoardo Mosca[*]
TU Munich,
Department of Informatics,
Germany
edoardo.mosca@tum.de

Georg Groh
TU Munich,
Department of Informatics,
Germany
grohg@in.tum.de

Abstract

State-of-the-art machine learning models are prone to adversarial attacks: Maliciously crafted inputs to fool the model into making a wrong prediction, often with high confidence. While defense strategies have been extensively explored in the computer vision domain, research in natural language processing still lacks techniques to make models resilient to adversarial text inputs. We adapt a technique from computer vision to detect word-level attacks targeting text classifiers. This method relies on training an adversarial detector leveraging Shapley additive explanations and outperforms the current state-of-the-art on two benchmarks. Furthermore, we prove the detector requires only a low amount of training samples and, in some cases, generalizes to different datasets without needing to retrain.

1 Introduction

Adversarial examples are slightly perturbed input samples purposely crafted to fool a target model (Szegedy et al., 2014). Despite being similar to the original samples, they are often misclassified with high confidence (Goodfellow et al., 2015). Without effective defense techniques, machine learning models become unusable in high-stakes situations and safety-critical tasks (Sharma et al., 2019).

Research in computer vision has extensively worked on better understanding adversarial image attacks and developing more robust models (Madry et al., 2018; Ozdag, 2018). However, the literature in *Natural Language Processing (NLP)* has witnessed fewer advances concerning this issue (Mozes et al., 2021; Zhou et al., 2019; Wang et al., 2019).

Text data needs to fulfill several properties such as lexical, grammatical, and semantic constraints. Thus, many efficient adversarial image attacks—e.g. gradient-based ones—are not transferable as they would lead to incorrect characters and non-existing terms (Zhang et al., 2020). However, word-level attacks that can preserve semantical information without introducing noticeable inconsistencies are particularly effective and not detectable via spell checkers (Garg and Ramakrishnan, 2020; Ren et al., 2019).

The lack of defense strategies against word-level text attacks motivates our research as this is a major obstacle to the safe deployment of NLP models. This work's contribution can be summarized as follows:

(1) Based on an analogous idea from computer vision (Fidel et al., 2020), we propose an adversarial attack detector leveraging *SHapley Additive exPlanations* (SHAP) to accurately recognize input manipulations (Lundberg and Lee, 2017). Results show that it outperforms the previous state of the art in adversarial detection on multiple datasets (Mozes et al., 2021).

(2) We analyze our method in terms of data efficiency and generalization. The proposed approach still offers competitive performance when trained on very little data and can even be transferred to unseen datasets while almost matching the previous state of the art.

[*] These authors contributed equally

Proceedings of the 7th Workshop on Representation Learning for NLP, pages 156 - 166
May 26, 2022 ©2022 Association for Computational Linguistics

(3) Alongside the quantitative analysis and its results, we visualize the space of generated Shapley-value-based explanations. This qualitative analysis sheds light on the reasons behind our method's high performance and desirable properties.

2 Related Work

2.1 Adversarial Text Attacks

An adversarial text attack is an artificial input obtained by modifying a sample from the available data. Normally, the altered text is similar—syntactically, semantically, or both—to the original one. However, their corresponding classification output substantially differs. Attacks can be either *targeted* or *untargeted* (Tao et al., 2018). Attacks of the first type aim to create misclassification results w.r.t. a specific class whereas the latter type wants to generate a misclassification regardless of the exact class.

Methods like DeepWordBug (Gao et al., 2018) or Hotflip (Ebrahimi et al., 2018) introduce character-level noise to create typos and grammatical inconsistencies in the sentence. These adversarial examples appear very similar to the original samples, but do not perfectly preserve their meaning and can be recognized due to their lexical incorrectness.

Other types of attacks instead alter the text at the word level and produce semantically equivalent and grammatically correct sentences to the initial input. Examples of techniques using this strategy are PWWS (Ren et al., 2019), TextFooler (Jin et al., 2020), and BAE (Garg and Ramakrishnan, 2020).

2.2 Defense Strategies for Computer Vision

Robustness against adversarial attacks—and especially their automatic detection—has been more exhaustively researched for computer vision applications rather than for text inputs. Hence, we briefly present a selection of the most promising approaches.

Xu et al. (2018) propose *Feature Squeezing*, based on the assumption that feature spaces are often unnecessarily large and leave extensive possibilities for an attacker to generate adversarial examples. Their approach leverages this fact by comparing the prediction of the original input image with a simplified one. When this difference surpasses a specific threshold, the input is classified as adversarial.

Roth et al. (2019) detect adversarial examples by measuring statistical differences between original and perturbed logits. According to their results, output logits corresponding to adversarial examples exhibit a much larger variation than normal samples when the input is perturbed.

Integrating explainability to detect adversarial examples has already been shown to be beneficial. Fidel et al. (2020) detect patterns in the SHAP signatures of input images (Lundberg and Lee, 2017). For normal samples, the inter-class SHAP signatures share common characteristics. For adversarial examples, however, the SHAP signatures show a mixture between two classes which can easily be detected using an additional classification model.

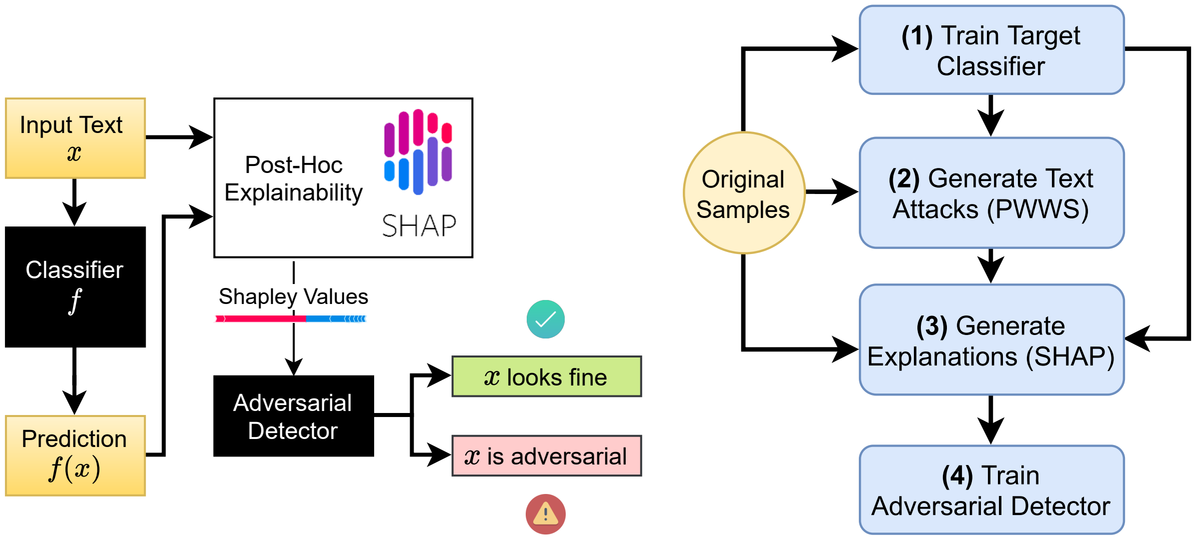
2.3 Defense Strategies for Natural Language Processing

Character-level attacks can be countered with defenses based on spell checkers (Pruthi et al., 2019; Huang et al., 2019). Nonetheless, those same defenses are extremely vulnerable to word-level attacks capable of preserving language coherence (Wang et al., 2019). Effective methods against syntactically correct attacks are *Adversarial Training* (AT) (Goodfellow et al., 2015), *Dirichlet Neighborhood Ensemble* (DNE) (Zhou et al., 2020), *Adversarial Sparse Convex Combination* (ASCC) (Dong et al., 2021) and *Synonym Encoding Method* (SEM) (Wang et al., 2019). The first three leverage some form of data augmentation to train the model on perturbed samples as well. The last, instead, introduces an encoder step before the target model's input layer and trains it to eliminate potential perturbations.

Particularly relevant for this work are *adversarial detection* methods. In contrast to other defenses, they can explicitly recognize manipulated inputs and send an alert signal. For natural language data, the available methods are *Frequency-Guided Word Substitution* (FGWS) (Mozes et al., 2021) and *learning to DIScriminate Perturbation* (DISP) (Zhou et al., 2019). The first—exploiting frequency properties of adversarial words—is the most recent and accurate method. Its authors showed medium to high F1 detection scores in a range from 62.2-91.4%, varying on the type of attack and target model.

2.4 Feature Relevance Explainability Methods

Among explainability techniques, *feature relevance* methods are often used to explain predictions pro-



(a) Goal Pipeline

(b) Construction Steps

Figure 1: Our detector for recognizing adversarial examples: the overall pipeline once the detector is trained (a) and the necessary steps in order to train it (b). While generating many adversarial attacks and explanations is required for training, the detector can then be simply "plugged in" and deployed together with the classifier f.

duced by black-box models (Arrieta et al., 2020; Mosca et al., 2021). Their goal is to attribute a relevance score to each input feature. Such value should quantify the effect that the feature has on the output, i.e. their contribution to the model's prediction (Wich et al., 2021).

Some of these methods rely on computing the gradient of the output w.r.t. the input features (Simonyan et al., 2014; Sundararajan et al., 2017). Others, such as LRP (Bach et al., 2015) and DeepLIFT (Shrikumar et al., 2017), are specifically designed for neural networks and follow the information flow in a backward fashion through the model's architecture. The procedure continues one layer at a time until the input features are reached. LIME (Ribeiro et al., 2016) explains black-box models via a local surrogate that approximates their behavior around a single instance. The surrogate can be then interpreted directly to estimate each feature's relevance.

Lundberg and Lee (2017) prove that several popular feature relevance methods—including LIME, LRP, and DeepLIFT—belong to a broader class of approaches: *additive feature relevance methods*. The authors propose a unified view of such methods that, combined with the game-theoretic concept of Shapley values (Shapley, 1952), constitutes the SHAP framework. SHAP-based explanations are covered more in detail in Section 3.2 as they represent a fundamental component of our proposed method.

3 Methodology

Our defense belongs to the adversarial detection category and is strongly inspired by the work of Fidel et al. (2020), which detects image-based adversarial attacks for computer vision models by using SHAP signatures. This work, instead, studies the application of this idea to text attacks for NLP classifiers. As sketched in Figure 1a, our goal pipeline consists of multiple stages. First, the input is fed to a classifier trained on the task-at-hand, which outputs a prediction. Shapley values are then computed w.r.t. the outcome and passed onto a machine-learning detector that predicts whether the sample is an adversarial attack. Note that our detector does not make any assumption on the classifier and is hence model-agnostic.

The classifier targeted by the attacks becomes considerably more robust when used in combination with the adversarial detector. To achieve our goal, we have to take several steps in order to train our detector. These steps—also summarized in Figure 1b for the reader—are described in detail in the next sections.

3.1 Crafting Adversarial Text Attacks

To train and test our detector, we choose to craft attacks semantically similar to the original input. This choice preserves lexical and grammatical co-

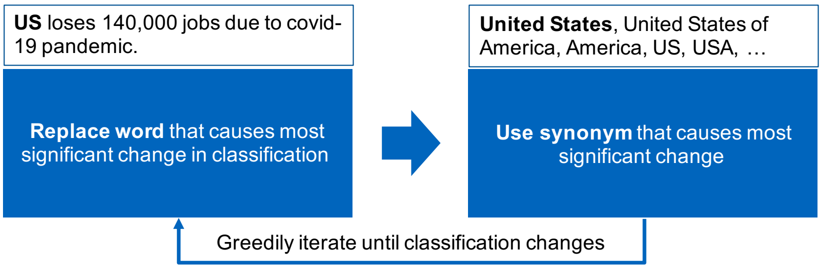

US loses 140,000 jobs due to covid-19 pandemic.
Replace word that causes most significant change in classification
United States, United States of America, America, US, USA, …
Use synonym that causes most significant change
Greedily iterate until classification changes

Figure 2: A simplified view of the generation of adversarial examples using PWWS (Ren et al., 2019)

herence also in adversarial sentences. We believe that such attacks are more subtle as they cannot be detected by spell checkers. In practice, for each sample x in the dataset, we generate

$$x^* = x + \Delta x, \|\Delta x\| < \epsilon \tag{1}$$

where Δx is a semantic perturbation and the classes predicted for x and x^* are different. To this end, we utilize the untargeted *Probability Weighted Word Saliency* (PWWS) method by Ren et al. (2019). This approach shows high effectiveness with good transferability. According to human evaluation, PWWS provides realistic examples with lexical correctness and only sporadic grammatical errors or semantic shifting (Ren et al., 2019).

The technique selects the word to be replaced based on two factors. The first is the change in the classification probability after substitution. The second, called *word saliency*, measures the variation in the output probability of the classifier if the word is set to unknown (out of vocabulary). The chosen word is then replaced by a word from a synonym set which causes the most significant change of classification probability. The algorithm greedily iterates until enough words have been replaced to change the final classification label. Figure 2 sketches the core idea behind the method.

3.2 Generating Model Explanations

Whenever classifying an input sentence as either regular or adversarial, our detector needs access to its corresponding feature relevance explanation. In other words, the detector takes its decision based on *how strong* each feature—in our case each word—influences the final model prediction. The assumption is that the model's reaction to original and adversarial samples is different even if the inputs look similar for a human. Thus, the model explanations for the two samples should also substantially differ from each other (Fidel et al., 2020).

We pick SHAP (Lundberg and Lee, 2017) to produce instance-level explanations to train the adversarial detector. This choice is motivated by the empirical superiority proven by its developers (Lundberg and Lee, 2017) and its previous successful applications in detecting attacks in computer vision. However, while Fidel et al. (2020) generate SHAP signatures w.r.t. the penultimate layer of the target model, we produce explanations directly w.r.t. the input sentence as text perturbations are introduced at the word level.

SHAP is based on a game theory concept—called Shapley values (Shapley, 1952)—originally used to fairly distribute a reward to a set of players that contributed to a certain outcome. In our case, the outcome is the model's prediction whereas the input features, i.e. the input words, are the players involved. Since the players most likely contributed differently to the turnout, their payout should differ based on their impact. Given a text classifier f and the set of all available features M, the Shapley value corresponding to each feature i is computed independently. More precisely, it is a weighted average of the relative outcome differences

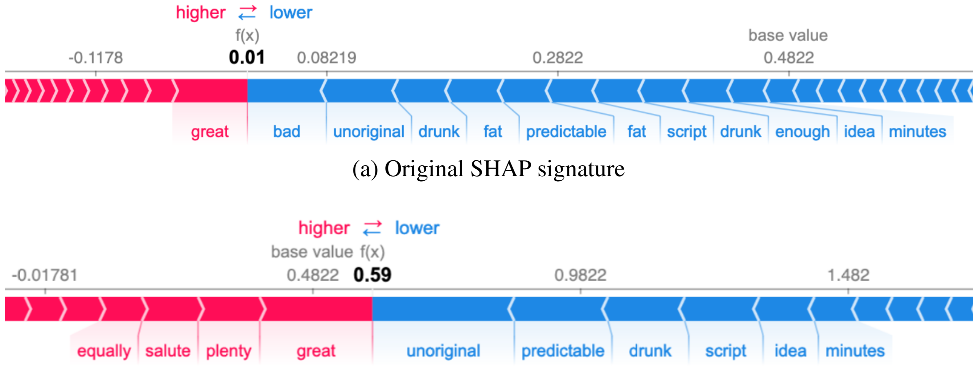
$$f(S \cup \{i\}) - f(S) \tag{2}$$

across all feature subsets $S \subseteq M \setminus \{i\}$.

As there are $2^{|M|}$ possible choices for S, exact Shapley values are exponentially complex to compute. However, the SHAP framework offers several methods to approximate them accurately and efficiently (Lundberg and Lee, 2017). In our work, we utilize DeepSHAP as it is tailored to deep learning models, which we utilize as targets for the text attacks (Lundberg and Lee, 2017). An official implementation has been made publicly available by the SHAP authors. [1]

Figure 3 shows two examples of explanations generated for *IMDb*, a movie review dataset (Maas

[1] https://github.com/slundberg/shap



(a) Original SHAP signature

(b) Adversarial SHAP signature

Figure 3: Force plots generated for a sample of the *IMDb* dataset and its corresponding adversarial attack. The *base value* indicates the average model's prediction across the whole dataset and $f(x)$ represents the model output probability for the selected instance. Red attributes drive the predictions towards class 1 (i.e. a positive review) and blue ones towards class 0 (i.e. a negative review). Starting from the base value (~ 0.48) and adding up all word contributions we reach the final prediction of 0.01. Hence, the original sample is classified as negative with high confidence. In the adversarial SHAP signature, most negative words were replaced by synonyms such that the prediction is now positive.

et al., 2011), with DeepSHAP. The first (Figure 3a) was generated from an original sample while the second (Figure 3b) from its corresponding adversarial attack generated with PWWS. As we can see, the attack changes substantially the effect that words have on the prediction. Hence, word-level contributions are a major indicator for detecting parts of a sentence that have a suspiciously high impact on the model decision. This supports our initial hypothesis that SHAP explanations do not rely on image-only properties and therefore can also serve as features for an adversarial detector in the NLP domain.

3.3 Target Model and Detector Architectures

Our pipeline includes two machine learning models: the text classifier trained for the task-at-hand and the adversarial detector.

For consistency with Mozes et al. (2021), used later for performance comparison, we chose a Bidirectional LSTM (Bi-LSTM) (Schuster and Paliwal, 1997) as architecture to be targeted by the adversarial attacks. However, other NLP models can also be utilized as the detector does not make any assumption on the classifier. The text inputs are first trimmed and padded to an equal length of 100. Increasing the input length drastically increases complexity along the pipeline while only yielding minor accuracy gains. Tokens are transformed into GloVe embeddings (Pennington et al., 2014) before being fed to the Bi-LSTM core layer. We attach a fully connected head layer to compute output probabilities. We adjust the number of output neurons based on the dataset currently in use.

SHAP values are extracted from the model for all output classes. Therefore, the SHAP signatures passed to the detector are numerical vectors of dimensionality [#classes × 100]. Here, each numerical value corresponds to the impact of a single word w.r.t. the model's output. We do not pick any particular architecture for our adversarial detector. Instead, we experiment with a variety of relatively simple machine learning models to test their performance. We include a *random forest* (Breiman, 2001), a *Support Vector Machine* (SVM) (Boser et al., 1992), and a simple two-layer-feed-forward neural network (Rumelhart et al., 1985).

3.4 Overall Pipeline and Experimental Setup

With the methodology for the main steps outlined in the previous sections, we now describe in greater detail how those steps are combined, following what we initially presented in Figure 1b. We repeat the procedure for each text dataset utilized for testing. These will be presented later in our evaluation section (4).

To begin with, we train the Bi-LSTM model on the given dataset. We consider this step concluded once the model converges to a satisfactory accuracy. This is usually around 90% accuracy, depending on the dataset. After that, we utilize PWWS as proposed by Ren et al. (2019)—implemented

	Method	AG_News	IMDb	SST-2	Yelp Polarity	Metric
Our	Neural Network	0.90 / 0.90	**0.96 / 0.96**	0.75 / 0.75	**0.94 / 0.94**	F1 score / Accuracy
	Random Forest	**0.91 / 0.91**	0.87 / 0.87	**0.77 / 0.77**	0.84 / 0.84	F1 score / Accuracy
	SVM	0.90 / 0.90	0.90 / 0.90	0.74 / 0.74	0.89 / 0.89	F1 score / Accuracy
SotA Detector	FGWS (Mozes et al., 2021)	-	0.77	0.63	-	F1 score
Other Defenses	DNE (Zhou et al., 2020)	**0.91**	0.82	-	-	Accuracy
	SEM (Wang et al., 2019)	0.76	0.85	-	-	Accuracy
	ASCC (Dong et al., 2021)	-	0.77	-	-	Accuracy

Table 1: Performance of different detector architectures on the *AG_News, IMDb, SST-2* and *Yelp Polarity* datasets. For comparison, we report also the defense performance of *Frequency-Guided Word Substitutions* (FGWS), *Dirichlet Neighbourhood Ensemble* (DNE), *Synonym Encoding Method* (SEM) and *Adversarial Sparse Convex Combinations* (ASCC).

in the TextAttack library [2]—to produce adversarial attacks targeting our trained NLP model. We generate one attack for each sample in the dataset. Instance-level explanations—i.e. Shapley value approximations—are then created via SHAP, both for normal and adversarial samples (Lundberg and Lee, 2017).

We combine all explanations to compose a balanced dataset for our adversarial detector. The data is split into training and test sets following an 80/20-ratio. We further used the default hyperparameters for all models in the framework. To allow for optimal reproducibility, we seeded all of our experiments. For the neural network-based detector, we pick layers of size 400 using a ReLU activation and an L1 weight regularizer to avoid overfitting. To further increase regularization, Dropout is used (Srivastava et al., 2014). The model is then trained for 10 epochs using the Adam optimizer with a learning rate of 0.001 and β_1, β_2 set to their default values of 0.9 and 0.99 respectively (Kingma and Ba, 2015).

4 Evaluation

4.1 Performance Results

We evaluate our approach on four major datasets often used in research, namely *IMDb* (Maas et al., 2011), *SST-2* (Socher et al., 2013), *Yelp Polarity* and *AG_News* (Zhang et al., 2015). While the last one classifies news articles into four distinct categories, the other three are binary sentiment analysis tasks on movie review data. The reviews are not fed into the detector directly but their corresponding SHAP signatures are instead. The number of samples in the datasets used for the experiment is reported in Table 2. Every dataset consists of a 50:50 split between original and adversarial samples and the sizes are varying between 940 (*Yelp Polarity*) and 100,000 (*AG_News*) samples.

Dataset	Size	#Normal	#Adversarial
AG_News	100,000	50,000	50,000
IMDb	3,580	1,790	1,790
SST-2	3,162	1,581	1,581
Yelp Polarity	940	470	470

Table 2: Sizes of the individual SHAP signature datasets used for training the adversarial detector. All datasets consist of 50% normal and 50% adversarial signatures.

Table 1 shows the performance of various detector architectures on the four datasets together alongside results achieved by previously proposed methods. To the best of our knowledge, the FGWS method proposed by Mozes et al. (2021) is the best detector currently available. With our SHAP-based classifiers, we significantly outperform their method on the *IMDb* dataset by 19% with an F1-score of 96% and on the *SST-2* dataset by 14% with an F1-score of 77%. Relatively simple machine learning models like a random forest or a support vector machine are able to classify the data very accurately. Both Mozes et al. (2021) and our work evaluate their defenses against PWWS targeting a Bi-LSTM model.

Besides adversarial detectors, we also outperform all other existing defenses to the best of our knowledge. On *IMDb*, our approach improves by 11% accuracy compared to the best method (Wang et al., 2019). On *AG_News*, it is matched only by the DNE method from Zhou et al. (2020). For each approach considered, we report the result w.r.t. the configuration achieving the best performance against PWWS from their corresponding original work. For completeness, we mention that Zhou et al. (2019) reports great results but their performance is not comparable as they do not test their method against any well-established attack.

[2] `https://github.com/QData/TextAttack`

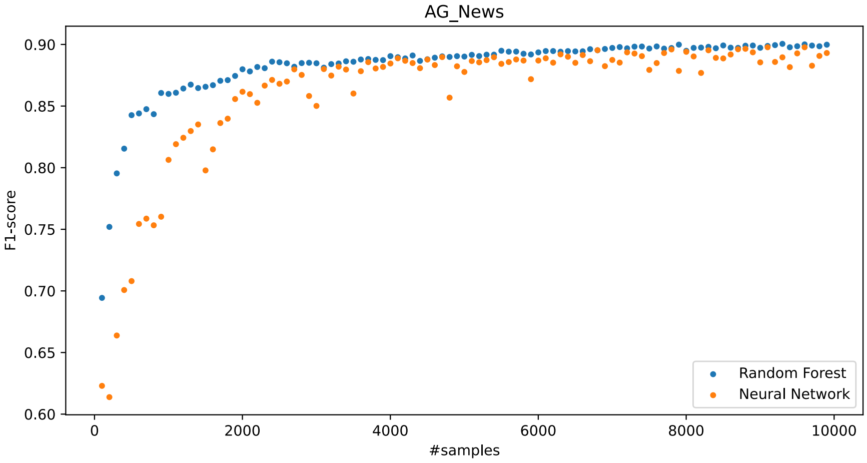


Figure 4: F1-scores for independent runs on the *AG_News* dataset using differently sized subsets of the training data. The F1-score starts to plateau after a few thousand samples for all detectors which shows data efficiency.

Classifier	Unnormalized SHAP	Unnorm. SHAP + Predicted Class	Normalized SHAP
Neural Network	0.90	0.90	0.90
Random Forest	0.91	0.91	0.92
SVM	0.90	0.90	0.90
Linear SVM	0.67	0.67	0.65

Table 3: F1-scores of input modifications for the detectors on the *AG_News* dataset.

To further improve the predictive performance of the model, we also included the predicted class coming from the base model as an input feature for the detector. As shown in Table 3, this had neither a positive nor a negative influence on the performance of the model. Normalizing the SHAP signatures only led to minor improvements for random forests and neural networks. This can be explained by the fact that all input features are Shapley values and are therefore in the same range.

4.2 Transferability

Base-Model	IMDb (Test)	SST-2 (Test)
IMDb	-	0.56
SST-2	0.42	-
Yelp Polarity	**0.71**	**0.66**

Table 4: F1-scores of the inference step with *IMDb* and *SST-2* datasets on neural network base-models which were trained on *IMDb, SST-2* and *Yelp Polarity*.

During our research the question arose whether the detectors are agnostic to the dataset or highly specialized. To evaluate this property, we trained three base-models with a neural network backbone on the *IMDb*, *SST-2* and *Yelp Polarity* datasets. Then, we performed the inference step with the *IMDb* and *SST-2* test sets on all three detectors and observed how the performance varies with different dataset combinations.

The results can be seen in Table 4. We report the strongest results when the detector was tested on the same dataset that was also used during training. This resulted in our competitive F1-scores of 94% on *IMDb* and 77% on *SST-2*. Interestingly, there existed other combinations which also produced results comparable to the state of the art, although the performance dropped compared to our strongest detectors. To be precise, the base-model which was trained on *Yelp Polarity* achieved good F1-scores on test sets of *IMDb* with 71.5% and of *SST-2* with 66%. In comparison, the state-of-the-art detector tested with similarly generated adversarial samples on a LSTM with PWWS by Mozes et al. (2021) achieved F1-scores of 77.4% on *IMDb* and of 63.4% on *SST-2*.

Such results are yet not strong enough to prove full generalization capabilities. However, we find them promising as they indicate that our detectors are in some cases actually transferable to other datasets once trained. Future research is crucial as in practice it allows to reuse models for different tasks.

4.3 Data efficiency

While our approach offers state-of-the-art detection performance of adversarial attacks, the corresponding detector model can be trained with a surprisingly low amount of data. To evaluate this property, we trained a neural network and a random forest on incremental subsets of the *IMDb* dataset where all runs were conducted independently from each other. We started with a dataset size of 100 and incrementally increased the number of samples up to 10,000. From Figure 4 one can directly observe the limited amount of data needed for the model to converge. For a neural network about 4,000 samples are needed before the F1-score starts to plateau. For a random forest classifier even less data is sufficient with around 3,000 samples.

4.4 Qualitative Results

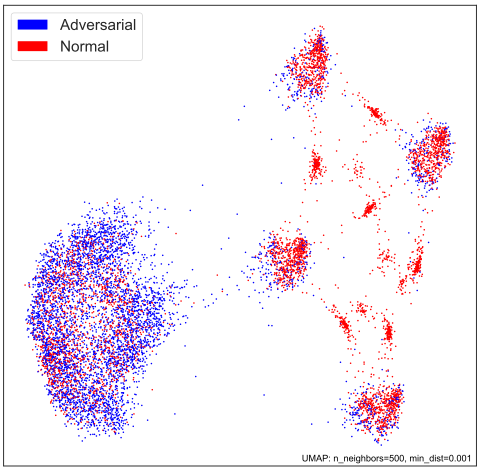


Figure 5: Visualization of the SHAP signatures of the *AG_News* dataset using UMAP. We randomly selected 10% of the samples to avoid overplotting.

In order to understand how the detector is able to distinguish between normal and adversarial inputs, we visualized the SHAP signatures in a two-dimensional space. To project the samples we rely on the UMAP dimensionality reduction algorithm proposed by McInnes et al. (2020). It is based on the fact that most high-dimensional data actually lies on a much lower-dimensional manifold and can be explained by a reduced number of variables. Figure 5 clearly shows four distinct red clusters corresponding to the four classes of the *AG_News* dataset. Regardless of their original class, most of the adversarial samples collapse into a single cluster which is clearly separable from the others. This explains why rather simple detector models are sufficient to accurately differentiate between normal and adversarial inputs. Our result is consistent with the experiments done by Fidel et al. (2020) which performed a similar analysis on SHAP signatures for images from the CIFAR-10 dataset (Krizhevsky et al., 2009).

4.5 Limitations

After the success in computer vision (Fidel et al., 2020), this work shows that SHAP values are also a valuable asset for discriminating between original and adversarial text samples. However, while word-level explanations are particularly effective at detecting word-level attacks, it is unclear how they would transfer to more sophisticated text manipulations. We believe this is a vulnerability as future attacks could involve using negations or paraphrasing whole sentences instead of unigrams.

While the approach's pipeline is intuitive and the results look promising, further research needs to study transferability to more complex target models such as transformers architectures. At the same time, we hope that future research also focuses on creating standard benchmarks to facilitate performance comparisons with previous defense methods.

5 Conclusion

Adversarial text examples are a major challenge for current research and represent an obstacle for safely deploying NLP models in high-stakes applications. While attacks are hard to be distinguished from their corresponding originals, patterns in the model's reaction can be recognized and leveraged using SHAP signatures for detecting manipulated input samples.

Our work trains a machine learning detector using SHAP explanations of normal and adversarial samples generated with PWWS. The proposed method is both intuitive and effective since it allows to detect parts of a sentence that have a suspiciously high impact on the model prediction and therefore distinguishes between regular and manipulated samples. Furthermore, our detector is model-agnostic as it does not make any assumption on the classifier targeted by the attacks.

Our approach achieves high accuracy and considerably outperforms the previous state of the art. In terms of data efficiency, we prove that the method can achieve nearly optimal performance also when using a small portion of the available data for training. A qualitative analysis of the SHAP signature landscape shows most adversarial samples contained in a single cluster, suggesting that model explanations explicitly encode information to separate attacks from their counterpart. We believe this result explains why relatively simple detector architectures suffice to achieve good performance results.

In terms of transferability to multiple datasets, our results are promising but yet not sufficient to prove full generalization capabilities. Although in some cases we match state-of-the-art performance even when training on one dataset and testing on another, our results are highly dependent on the dataset pair.

We encourage future research to continue working on generalization across multiple data sources and to evaluate performance against multiple types of attacks and models. We believe our contribution can help researchers to develop better defense strategies against attacks and thus promoting the safe deployment of NLP models in practice. We release our code to the public to facilitate further research and development [3].

References

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):130–140.

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2021. Towards robustness against natural language word substitutions. In *9th International Conference on Learning Representations (ICLR)*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Gil Fidel, Ron Bitton, and Asaf Shabtai. 2020. When explainability meets adversarial learning: Detecting adversarial examples using SHAP signatures. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. Achieving verified robustness to symbol substitutions via interval bound propagation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4083–4093.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 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*, volume 34, pages 8018–8025.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Alex Krizhevsky et al. 2009. Learning multiple layers of features from tiny images.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

[3] https://github.com/huberl/adversarial_shap_detect_Repl4NLP

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction.

Edoardo Mosca, Maximilian Wich, and Georg Groh. 2021. Understanding and interpreting the impact of user context in hate speech detection. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 91–102.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 171–186, Online. Association for Computational Linguistics.

Mesut Ozdag. 2018. Adversarial attacks and defenses against deep neural networks: A survey. *Procedia Computer Science*, 140:152–161. Cyber Physical Systems and Deep Learning Chicago, Illinois November 5-7, 2018.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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*, pages 1135–1144.

Kevin Roth, Yannic Kilcher, and Thomas Hofmann. 2019. The odds are odd: A statistical test for detecting adversarial examples. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5498–5507. PMLR.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Lloyd S. Shapley. 1952. *A Value for n-Person Games*. RAND Corporation, Santa Monica, CA.

P. Sharma, D. Austin, and H. Liu. 2019. Attacks on machine learning: Adversarial examples in connected and autonomous vehicles. In *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–7.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Guanhong Tao, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang. 2018. Attacks meet interpretability: Attribute-steered detection of adversarial samples. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Xiaosen Wang, Hao Jin, and Kun He. 2019. Natural language adversarial attacks and defenses in word level. *arXiv preprint arXiv:1909.06723*.

Maximilian Wich, Edoardo Mosca, Adrian Gorniak, Johannes Hingerl, and Georg Groh. 2021. Explainable abusive language classification leveraging user and network data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 481–496. Springer.

Weilin Xu, David Evans, and Yanjun Qi. 2018. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Proceedings 2018 Network and Distributed System Security Symposium*. Internet Society.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing. *ACM Transactions on Intelligent Systems and Technology*, 11(3):1–41.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA. MIT Press.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-wei Chang, and Xuanjing Huang. 2020. Defense against adversarial attacks in nlp via dirichlet neighborhood ensemble. *arXiv preprint arXiv:2006.11627*.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4904–4913, Hong Kong, China. Association for Computational Linguistics.