ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Adversarials^{-1} in Speech Recognition: Detection and Defence

Nils Worzyk, Stefan Niewerth, and Oliver Kramer *

University of Oldenburg - Dept. of Computing Science
26129 Oldenburg - Germany

Abstract. Systems that accept voice commands have become established in our daily lives. To process those commands, modern systems usually use neural networks, which have been shown to be very successful. Nevertheless, they are vulnerable against adversarial attacks—slightly perturbed inputs, to fool the system, but are not recognizable by humans. In this work we extend the adversarial^{-1} concept, introduced in the image domain, to the speech recognition domain. By adapting the methodology we are able to identify adversarial inputs, in certain cases, with an accuracy of 99.9%, while still detecting benign inputs with an accuracy of 99.8%, for the investigated attacks. Furthermore, we present a technique to restore the correct label of an adversarial input, with up to 67.6% accuracy. All program code for this work can be found on `https://github.com/OLStefan/Adversarials-1Speech-Recognition`.

1 Introduction

Over the last few years, voice-controlled systems have become increasingly wide spread. Many of those systems incorporate neural networks to process speech data and understand the given commands. One example for such a system is the DeepSpeech project by the Mozilla Foundation, which is based on research by Hannun et al. [1]. Since that project is openly accessible[1], it is also used in this paper.

At the same time, it has been shown that neural networks are vulnerable against *adversarial inputs* [2]—slightly perturbed inputs, where the manipulation is unnoticeable for humans, but leads the artificial system to misinterpret the input. More recently, adversarial attacks have also been shown to be applicable to speech recognition systems, which could lead to misuse. Carlini and Wagner for example propsed a white-box attack [3], i.e., they assume knowledge about the gradient of the model to attack. In contrast, Alzantot et al. [4] propose a black-box attack, i.e., they assume an adversary does not have knowledge about the model, but only can query it to get information. Both attacks are used in this paper.

In order to defend against those attacks, prior works proposed to use more sophisticated preprocessing of the inputs, e.g., quantization, local smoothing, or down-sampling. Rajaratnam et al. [5] also propose compression and filtering

*This research is funded by the German Research Foundation through the Research Training Group DFG-GRK 1765: "System Correctness under Adverse Conditions".

[1] https://github.com/mozilla/DeepSpeech

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

techniques, as well as using ensembles to identify adversarial inputs. In their work, they reported a maximal F1-score of 0.91 when using compression/filter techniques, resp. a maximal F1-score of 0.92 when using ensembles. Another recent work of Zeng et al. [6] also use ensembles of different automatic speech recognition systems, and, based on the similarities of the outputs, differentiate between benign and adversarial inputs. They report a detection accuracy of 99.78%

In this work, we transfer a defending technique introduced by Worzyk & Kramer [7] from the image classification domain to speech recognition. The basic idea is to attack an unknown input with an internal, known attack, and measure the difference between the input, and the internally manipulated counterpart. This process will be explained in more detail in Section 3. Thereby, we are able to detect 99.9% of adversarial inputs as adversarial, as well as 99.8% of the original inputs as benign. In addition to a very high detection rate of adversarial inputs, we adapted the process to restore the original class of adversarial inputs. With the adaptation, we are able to correctly classify adversarial inputs with an accuracy of 67.6%.

The remainder of this paper is structured as follows. In Section 2 we will give a very brief introduction to the speech model "DeepSpeach" used in this paper. Our defence is split into a detection phase (Section 3) and a restoration/classification phase (Section 4). The corresponding results will be given in the separate chapters. In Section 5 we will conclude our paper. All program code for this work can be found on `https://github.com/OLStefan/Adversarials-1Speech-Recognition`.

2 DeepSpeech

DeepSpeech is an open source[2] speech recognition project, based on the work of Hannun et al. [1], and works in three steps. 1.) During *Preprocessing*, the Mel-frequency cepstrum coefficients (MFCC) of the audio signal are calculated. Those coefficients resemble frequencies, which are important in human hearing. 2.) The MFCC are fed into the model, a bidiractional neural network to calculate a "sequence of character probabilities for the transcription y, with $\hat{y}_t = \mathbb{P}(c_t|\mathcal{X})$, where $c_t \in \{a, b, c, \ldots, z, \textit{space}, \textit{apostrophe}, \textit{blank}\}$" [1]. As a loss function to train the network connectionist temporal classification (CTC) [8] is used. 3.) In a *post-processing* step, the predicted phase is compared with a language model. Trained on the Speech Command Dataset [9], this model achieves an accuracy of 83.86%.

3 Detection

The basic idea for this paper was introduced by Worzyk & Kramer [7] for the domain of image classification. They proposed a two-step defence technique,

[2]https://github.com/mozilla/DeepSpeech

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

where the first step is to detect adversarial inputs, and in a second step, they try to reverse the attack and restore the correct class of a manipulated input.

The adapted workflow to detect adversarial inputs is depicted in Figure 1. After receiving an unknown input, we use a known attack to create a manipulated version of the input. Because the defender has knowledge about the system, he is free to use black-box attacks, as well as white-box attacks. If the unknown input was benign, the internal counterpart is an adversarial input, while, if the input was already an adversarial input, the internal counterpart is called adversarial^{-1}. Based on the differences between the unknown input and the internal counterpart, previously trained classifiers predict, whether the unknown input was benign or adversarial.

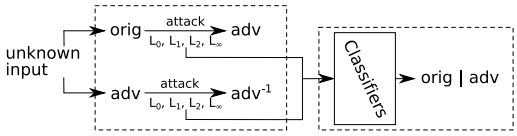

Fig. 1: Workflow of the detection stage. The unknown input is attacked. The L_0, L_1, L_2 and L_∞-differences between the input and manipulated version are used to predict, whether the unknown input was original or adversarial.

In this paper, we use the attacks proposed by Alzantot et al. [4] and Carlini & Wagner [3], to produce the adversarial inputs, as well as the internal counterparts. Especially in the later described classification stage (cf. Section 4) the randomness of Alzantot's attack showed to be very usefull. After the attack, we calculate the L_0, L_2, and L_∞ norm of the input audio file $\vec{x}$ and the manipulated audio file $\vec{x}'$, as well as the L_1, L_2, and L_∞ norm of the input MFCC values and the manipulated MFCC values. Afterwards, the differences of the norms are calculated. This results in six different parameters, i.e., three differences between the raw audio files, and three differences between the MFCC values. Based on these values the input is classified as benign or adversarial.

The classifiers we used were k-nearest Neighbours (kNN) with $k = 10$, a Decision Tree (DT) classifier, and a simple MLP with three layers. The results for the MLP are omitted, because the are way worse than the results of the other two internal classifiers. All classifier implementations are based on the machine learning library scikit-learn [10].

For our experiments, we used a subset of the SpeechCommand dataset [9], and concentrated on the ten words: “down”, “go”, “left”, “no”, “off”, “on”, “right”, “stop”, “up”, “yes”. For each of the words, 175 samples were randomly chosen and manipulated to each of the other nine possible words, resulting in 15,750 adversarial inputs. Since we are using 2 different attacks, in total 31,500 adversarial samples, resp. samples of differences between original and adversarial inputs were created.

Of those one time manipulated words, we chose 225 random samples of each word, manipulated by each attack in the first step, and attacked them again with the internal attack. Hence, we created 81,000 adversarial^{-1} inputs. In total, we

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

thereby created a dataset of 112,500 samples, consisting of differences between original and adversarial inputs, resp. between adversarial and adversarial^{-1} inputs. The whole dataset was spilt into a training set consisting of 90%, i.e., 101,250 samples, and a test set consisting of the remaining 10%, i.e., 11,250 samples. Furthermore, to increase statistical evidence, we repeated our training and testing 50 times, with each time a new randomly shuffled dataset.

The results for all attack combinations and classifiers are shown in Table 1. Column *attack 1* indicates the attack used by the adversary, while column *attack 2* indicates the attack used as internal attack. The rows indicate the subset of the whole dataset, on which the experiments were performed.

Table 1: Detection accuracy of adversarial (specificity) and benign (sensitivity) inputs.

attack 1	attack 2	classifier	sensitivity	specificity	accuracy
Alzantot	Alzantot	kNN	89.7%	99.7%	94.70%
		DT	99.8%	99.9%	99.85%
	Carlini	kNN	97.5%	98.3%	97.90%
		DT	**99.9**%	97.7%	98.80%
Carlini	Alzantot	kNN	88.4%	100%	94.20%
		DT	99.8%	99.9%	**99.85**%
	Carlini	kNN	92.8%	99.4%	96.10%
		DT	94.2%	**100.0**%	97.10%

Overall, it can be seen, that DT as internal classifier yield the best results, if we use the attack of Alzantot et al. [4] as internal attack. Thereby, we achieve the best mean sensitivity of 99.8%, as well as the best specificity of 99.9%.

4 Classification

In the classification stage, Worzyk & Kramer [7] proposed to use an untargeted internal attack, to push an adversarial input over the nearest decision boundary, towards the assumed to be most likely original class. However, in audio classification, the attacks proposed so far are all targeted attacks. Thereby, we had to adapt their idea to serve our purposes.

The technique, which worked in the end was to extend the classifiers used in the detection stage, to also predict the original class. Thereby we found, that when the target of the internal attack was the same, as the original correct class, the prediction accuracy of the correct class was very high. In contrast, if the target of the second attack was something different than the original correct class, the prediction accuracy was more or less uniform among the possible classes. In Figure 2 these distribution differences are shown. Considering all original "left" inputs, which were manipulated to all the other classes, if the target of the internal attack is left as well, the predicted original class is also "left" for 94.8% of the inputs. However, if the target of the internal attack is "down", the prediction is more or less random among all possible classes.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

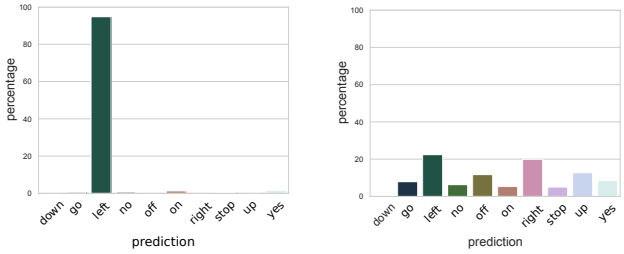

Fig. 2: Exemplary depiction of the different prediction distributions. Inputs of original class "left" are manipulated to all other classes. If the target of the internal attack is "left" (left figure) as well, the classifier predicts class left for many of the adversarial inputs. In contrast, if the target of the internal attack is "down", the predicted class is random among all possible classes.

Table 2: Correct label prediction of adversarial inputs.

	kNN	DT
accuracy	**67.6%**	33.6%

Based on this observation, we utilised the randomness of Alzantot's attack, to create 10 versions of each possible class for the unknown input and predicted the corresponding original class. Thereby, we get a distribution of predictions similar to Figure 2. Afterwards, we counted the number of predicted classes and made a majority vote to determine the final prediction for an unknown, but already identified adversarial input. The results for the decision tree (DT) as well as k-nearest Neighbours as internal classifiers are shown in Table 2. In contrast to the classification stage, kNN achieves the best results, with a correct classification accuracy of 67.6% for adversarial inputs.

5 Conclusion

We presented a defence to adversarial attacks in the domain of speech recognition. The defence itself is based on the work of Worzyk & Kramer [7], and composed of two stages, where the first is to detect adversarial inputs, and the second is to restore the original label.

In the first stage, we attack the given input with a given attack, to create a manipulated version. Based on differences, i.e., the L_0, L_2, and L_∞ differences between the raw audio files and L_1, L_2, and L_∞ differences between the Mel Frequency Cepstral Coefficients (MFCC), we trained different classifiers to distinguish between benign and adversarial inputs. This results into a classification accuracy of 99.85%, where the sensitivity, i.e., correct classification of benign inputs is 99.8%, and a specificity, i.e., correct classification of adversarial inputs is 99.9%. This result itself is better, than other recent methods, e.g. Rajaratnam

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

et al. [5] report a maximal F1-score of 0.91, when using compression methods, resp. 0.924 when using ensemble methods, or Zeng et al. [6] report an accuracy rate of 99.78%, when using ensembles of automatic speech recognition systems, and compare the output of the ensemble members.

In the second stage, we trained the classifiers not only to distinguish between benign and adversarial inputs, but to classify for the original labels. We observed a significantly higher classification accuracy, when the target class of the internal attack is the same as the original class, in comparison to different target and original classes. This behaviour was used to restore the original class successfully in 67.6%, on an independent test set of adversarial inputs.

In future work we consider to investigate the distributions even further. This might also be necessary in other speech recognition tasks with a larger corpus. One option could be, to randomly choose words, until we find an explicit distribution like in Figure 2.

References

[1] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

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

[3] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2018.

[4] Moustafa Alzantot, Bharathan Balaji, and Mani Srivastava. Did you hear that? adversarial examples against automatic speech recognition. In *2017 NIPS Machine Deception workshop*, 2017.

[5] Krishan Rajaratnam, Kunal Shah, and Jugal Kalita. Isolated and ensemble audio preprocessing methods for detecting adversarial examples against automatic speech recognition. In *Proceedings of the 30th Conference on Computational Linguistics and Speech Processing (ROCLING 2018)*, pages 16–30, 2018.

[6] Qiang Zeng, Jianhai Su, Chenglong Fu, Golam Kayas, and Lannan Luo. A multiversion programming inspired approach to detecting audio adversarial examples. In *The AAAI-19 Workshop on Artificial Intelligence for Cyber Security (AICS)*, 2019.

[7] Nils Worzyk and Oliver Kramer. Adversarials-1: Defending by attacking. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[8] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *International conference on Machine learning (ICML)*, pages 369–376. ACM, 2006.

[9] Pete Warden. Speech commands: A public dataset for single-word speech recognition. *Dataset available from http://download. tensorflow. org/data/speech_commands_v0*, 1, 2017.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.