Study on the Robustness of Multi-modal Models in Deception Detection

SPML final project

Gi-Luen Huang, Yu-Chieh Chao
Graduate Institute of Communication Engineering
{r09942171, r09942074}@ntu.edu.tw
Group 6

Abstract

Many researchers are now studying how to increase the accuracy of deception detection. The accuracy and the area under the curve (AUC) nearly achieved 100%. Recent work has demonstrated that deep neural networks are vulnerable to adversarial examples. However, the adversarial attack on deception detection has not been studied yet, and the robustness of the model should be particularly important for deception detection. In this work, we design the server and attacker model and conduct extensive experiments to investigate whether the adversarial examples will decrease the model's performance. After experiments, we find that the adversarial examples will significantly decrease the model's performance, especially the multi-modal attack, making the model's accuracy worse. Furthermore, even with adversarial examples, we find that adversarial training can effectively improve the robustness of model. The code is available at: https://github.com/come880412/Adversarial_attack_on_deception_detection.

1 Introduction

Deception detection is used to determine a person's truthfulness and credibility. It contains behavioral and physiological cues, from gestures to facial emotions to pose, which made researchers study how to recognize if someone is lying or not. The sign of deceit may not be single, it should consider a mixture of different nonverbal communications. Nonverbal communication of emotion can be observed from face to body and other physiological responses. A large amount of research showed that people's ability to recognize lying is not accurate, including all kinds of people, such as students, psychologists, judges, interviewees, and law enforcement officer. When it comes to putting criminals under arrest rather than innocents, the ability to precisely identify whether someone is lying or not is vital to the police officers. The traditional way of identification usually uses polygraph and fMRI, which has two disadvantages - a lot of information about signs of one's physiological changes and expensive equipment are required. Therefore, some papers explore deception detection by using facial or audio information recently. P'erez-Rosas *et al.* [9, 10] extracts features from gesture, expressions and linguistic models. Jaiswal *et al.*[6] applied visual and verbal cues to test deceit or truth prediction. Wu *et al.* [12] studied the importance of vision, audio, and text through sequential frames. They combined both micro-expressions of the human faces and the audio of the speakers. Ding *et al.* [2] identified whether a person is lying by how his gestures, poses change and facial emotion. However, although many researchers are now studying how to increase the accuracy of deception detection, no one has studied adversarial attack-related issues.

Preprint. Under review.

2 Related Work

2.1 Fast Gradient Sign Method(FGSM)

Goodfellow et al. [5] proposed the FGSM for generating adversarial examples. This method uses the gradient of the loss function of the model pertaining to the input feature vector. Given the input data, FGSM is to perturb the gradient direction of each feature by the gradient of the input data. After that, if you feed the perturbed data into the classifier, the classification result will be changed. For a neural network with cross-entropy loss function $C(X, y)$ where X is the input data and y is the target class for the input data. Then the adversarial examples are generated by the following formulation:

$$X_{adv} = X - \epsilon sign(\nabla_X C(X, y_i))$$

where ϵ is a parameter to determine the perturbation size.

2.2 Iterative Fast Gradient Sign Method(I-FGSM)

The I-FGSM is the iterative version of FGSM. This method applies FGSM many times with a small perturbation size instead of applying adversarial noise with one large perturbation size. The adversarial examples are generated by the following formulation:

$$X_{adv}^{N+1} = Clip_{X,\epsilon}\{X_{adv}^{N} - \alpha sign(\nabla_X C(X_{adv}^{N}, y_i)\}$$

where $Clip_{X,\epsilon}$ represents a clipping of the values of the adversarial example. This method has been proven that it is the most powerful method in the first-order method [7].

2.3 Momentum Iterative Fast Gradient Sign Method(MI-FGSM)

The MI-FGSM [3] is the I-FGSM with momentum factor. Beyond iterative gradient-based methods that iteratively perturb the input with the gradients to maximize the loss function, momentum-based methods accumulate a velocity vector in the gradient direction of the loss function across iterations, for the purpose of stabilizing update directions and escaping from poor local maxima. The adversarial examples are generated by the following formulation:

$$X_{adv}^{t+1} = X_{adv}^{t} - \alpha sign(g_{t+1})$$

where g_{t+1} can be defined as follow:

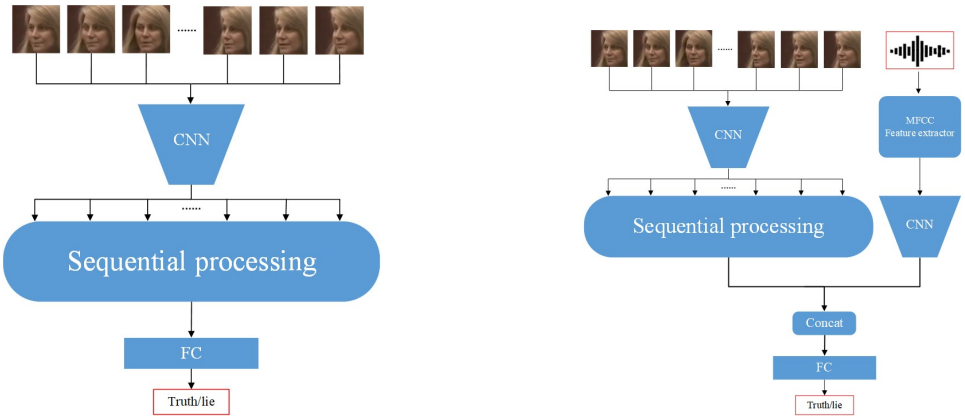
$$g_{t+1} = \mu \cdot g_t + \frac{C(x_t, y)}{\|\nabla_x C(x_t, y)\|_1}$$

2.4 Mel-Frequency Cepstral Coefficients (MFCC)

MFCC [8] is widely applied on the speech recognition system. Since the sound signal is continuously changing, we assume that the sound signal is fixed in a short time for simplification. First, take the Fourier transform of a sound signal and pass the transformed sound signal into the mel filter to obtain the mel scale. Secondly, take the logs of the powers at each of the mel frequencies and conduct the discrete cosine transform to transform the list of mel log powers into the cepstral domain. MFCC is the amplitudes of the cepstral, which typically uses 12 coefficients for representation. The figure of MFCC is illustrated in Figure 2.

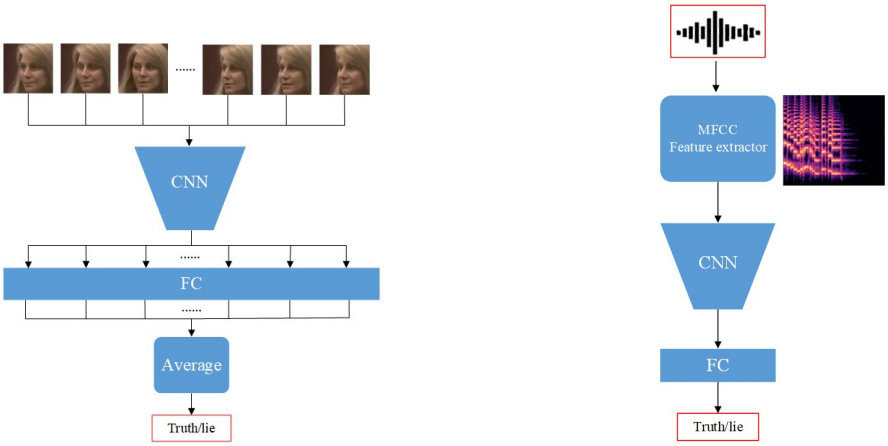
2.5 Sequential processing models

Recently, there are many papers that proposed many methods to process the sequential inputs, such as Transformer [11] or GRU [1]. Transformer-based models are widely used in the Natural Language Processing (NLP) field and are proved more accurate than RNN-based models. Therefore, some papers explore the transformer model into the computer vision (CV) field [4]. In this work, we employ the two types of models as our server model: CNN_GRU and CNN_Transformer.



(a) (b)

Figure 1: Server models. (a) Single-modal (b) Multi-modal



(a) (b)

Figure 2: Attacker models. (a) Video model (b) Audio model

3 Method

3.1 Server model

Figure 1 illustrates the framework of our server model. Our server model has single-modal and multi-modal. The former only takes the face images, and the latter takes both face images and audios into the model. In the design of the single-modal model, the CNN module embeds each face image into a feature vector. Such feature vectors proceed through the sequential processing module to obtain the representation of the sequential frames. Finally, the FC-layer determines whether the sequential inputs are lying or not. In addition, the multi-modal model takes both face images and audios into the model, fusing the frame vectors and audio vectors to make the final decision. For the server model, we select ImageNet-pretrained ResNet18 and ResNeXt50-32x4d as our CNN feature extractor.

3.2 Attacker model

The attacker's model architectures which is designed to generate adversarial examples are shown in Figure 2. Based on the experience gained from the gray-box attack homework, vector representations generated by relatively simple models are more general, which perform better on transfer attack; hence, we design the model as simple as possible. The video model (as shown in Figure 2 (a)) and the audio model (as shown in Figure 2 (b)) are then used for generating adversarial examples of video frame and audio data, respectively.

Table 1: Detection accuracy (%) of server models on standard Real-life dataset.

Input data	ResNet18 GRU	ResNet18 Transformer	ResNeXt50-32x4d GRU	ResNeXt50-32x4d Transformer
Video	78.18	**90.91**	81.82	86.36
Video + Audio	**95.45**	90.91	-	-

Table 2: Detection accuracy (%) of attacker models on standard Real-life dataset.

Input data	AlexNet	VGG16	ResNet18	ResNet50	ResNeXt50-32x4d
Video	72.73	89.10	84.55	**90.91**	**90.91**
Audio	72.73	72.73	81.82	90.91	**100.00**

For video model, video frames are sequentially input to the CNN layer, which generate vector representations for each frame. Then, these vector representations are fed into the classifier to make the truth/lie decision for each video frame. Finally, the binary decisions are averaged to get the final decision. And for the audio model, audio data is first transformed by the MFCC feature extractor then fed into the CNN layer. The extracted feature vector is then input into the fully-connected layer to perform deception detection. For the attacker model, we select ImageNet-pretrained AlexNet, VGG16, ResNet18, ResNet50, and ResNeXt50-32x4d as our CNN feature extractor.

3.3 Generate adversarial examples

To generate adversarial examples for both video frame and audio data, the aforementioned video model, audio model, and attack algorithms including FGSM, I-FGSM, and MI-FGSM are used. These attack algorithms are selected also due to the experience learned from the gray-box attack homework, where simpler attack algorithms have the better ability on transfer attack. For the video data, adversarial examples are generated by adding the identical perturbation to frames of the input video.

4 Experiments

4.1 Performance analysis

The experiments in table 1 and 2 demonstrate that the designed model architectures of both server and attacker achieve outstanding performance on the standard Real-life dataset. With the well-performing models, we then start examining the robustness of models on the server when being attacked by the adversarial examples generated by the attacker.

Table 3 demonstrates the deception detection accuracy of server video models when video frames are being attacked by the listed methods. Note that the Ensemble model of the attacker means that

Table 3: Detection accuracy (%) of video models on the server when video frame data are being attacked. Each method listed in the left-most column represents the adversarial video frames generated by the attacker model and the attack algorithm. While the first row represents server models.

	ResNet18 GRU	ResNet18 Transformer	ResNeXt50-32x4d GRU	ResNeXt50-32x4d Transformer
Original data	78.18	90.91	81.82	86.36
ResNet18 (FGSM)	71.82	63.64	48.18	63.64
Ensemble (FGSM)	67.27	63.64	**46.36**	63.64
ResNet18 (I-FGSM)	68.18	81.82	71.82	72.73
Ensemble (I-FGSM)	74.55	63.64	73.64	51.82
ResNet18 (MI-FGSM)	70.91	55.45	64.55	60.91
Ensemble (MI-FGSM)	**66.36**	**45.45**	61.82	**49.09**

Table 4: Detection accuracy (%) of multi-modal models on the server when video frame data are being attacked. Each name listed in the left-most column is the server model. While the first row represents the adversarial examples generated by the attacker model and the attack algorithm.

	Standard dataset	ResNet18 (FGSM)	Ensemble (FGSM)	ResNet18 (I-FGSM)	Ensemble (I-FGSM)	ResNet18 (MI-FGSM)	Ensemble (MI-FGSM)
ResNet18 GRU	95.45	92.73	88.18	93.64	79.09	92.73	**70.00**
ResNet18 Transformer	90.91	87.27	80.00	90.00	70.91	82.73	**67.27**

Table 5: Detection accuracy (%) of multi-modal models on the server when audio data are being attacked. Each name listed in the left-most column is the server model. While the first row represents the adversarial examples generated by the attacker model and the attack algorithm.

	Standard dataset	ResNet18 (FGSM)	Ensemble (FGSM)	ResNet18 (I-FGSM)	Ensemble (I-FGSM)	ResNet18 (MI-FGSM)	Ensemble (MI-FGSM)
ResNet18 GRU	95.45	94.55	95.45	91.82	91.82	**90.91**	92.73
ResNet18 Transformer	90.91	90.00	90.91	90.00	90.91	**90.00**	90.91

Table 6: Detection accuracy (%) of multi-modal models on the server when both video and audio data are being attacked. Each name listed in the left-most column is the server model. While the first row represents the adversarial examples generated by the attacker model and the attack algorithm.

	Standard dataset	ResNet18 (FGSM)	Ensemble (FGSM)	ResNet18 (I-FGSM)	Ensemble (I-FGSM)	ResNet18 (MI-FGSM)	Ensemble (MI-FGSM)
ResNet18 GRU	95.45	90.00	82.73	87.27	73.64	82.73	**68.18**
ResNet18 Transformer	90.91	80.91	80.00	88.18	63.64	66.36	**55.45**

Table 7: Detection robustness (%) of adversarial-trained video models on the server when video frame data are being attacked. Note that the adversarial training here is executed using the adversarial examples generated by ResNet18 (FGSM) method. Each method listed in the left-most column represents the adversarial video frames generated by the attacker model and the attack algorithm. While the first row represents the server model.

	ResNet18 GRU	ResNet18 Transformer	ResNeXt50-32x4d GRU	ResNeXt50-32x4d Transformer
Original data	89.09	88.18	90.91	87.27
ResNet18 (FGSM)	**90.00**	**89.09**	**90.91**	**81.82**
Ensemble (FGSM)	83.64	88.18	88.18	77.27

Table 8: Detection robustness (%) of adversarial-trained video models on the server when video frame data are being attacked. Note that the adversarial training here is executed using the adversarial examples generated by Ensemble (FGSM) method. Each method listed in the left-most column represents the adversarial video frames generated by the attacker model and the attack algorithm. While the first row represents the server model.

	ResNet18 GRU	ResNet18 Transformer	ResNeXt50-32x4d GRU	ResNeXt50-32x4d Transformer
Original data	81.82	79.09	79.09	81.82
ResNet18 (FGSM)	75.45	79.09	78.18	80.91
Ensemble (FGSM)	**78.18**	**85.45**	**79.09**	**81.82**

the CNN feature extractor of the model consists of AlexNet, VGG16, ResNet18, ResNet50, and ResNeXt50-32x4d. From the table, we can discover that, in most cases, the ensemble model with MI-FGSM algorithm achieves the best attacking result. Next, we are going to explore the robustness of multi-modal models. Experimental results in table 4, 5, and 6 show the robustness of multi-modal models when video frames, audio data, and both of them are attacked, respectively. Again, we can notice that the Ensemble model with MI-FGSM achieves the best attacking result in most situations. For instance, ResNet18-Transformer model only has the detection accuracy of 55.45 % when attacked by adversarial examples generated by Ensemble (MI-FGSM) method. Further, by comparing table 4 and 5, we find that the detection accuracy of the server model degrades more when video data are attacked. Hence, we assume that server models rely more on the video data to determine whether criminals are lying.

From the experiments above, we point out that adversarial attack is also a serious problem on deception detection task. With this, we attempt to proof whether adversarial training can improve the robustness of the deception detection models. We study the robustness of video models, which are adversarial trained. Results are demonstrated in table 7 and 8. Meanwhile, the adversarial training in table 7 and 8 are executed on the adversarial examples generated by ResNet18 (FGSM) and Ensemble (FGSM) method, respectively. From the experiments, we show that adversarial-trained models can actually become more robust. Moreover, adversarial-trained models are relatively robust on the corresponding attack.

5 Conclusion

Deception detection, as a security-related task, we have demonstrated the existence of adversarial attack in it. First of all, single-modal models, which consider either video frames or audio data as the input information, are both susceptible to adversarial examples. Then, we also show the detection degradation of the multi-modal models when video data, audio data, or both of them are being perturbed. Meanwhile, we find that the deception detection models rely more on video data to judge whether the criminal is lying. Finally, experiments indicate that the robustness of the deception detection model can be further improved via adversarial training. For the future work, the robustness of many state-of-the-art deception detection models can be further studied. And also, many large-scale datasets can be considered.

References

[1] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[2] Mingyu Ding, An Zhao, Zhiwu Lu, Tao Xiang, and Ji-Rong Wen. Face-focused cross-stream network for deception detection in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7802–7811, 2019.

[3] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[6] Mimansa Jaiswal, Sairam Tabibu, and Rajiv Bajpai. The truth and nothing but the truth: Multimodal analysis for deception detection. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 938–943. IEEE, 2016.

[7] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[8] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*, 2010.

[9] Verónica Pérez-Rosas, Mohamed Abouelenien, Rada Mihalcea, and Mihai Burzo. Deception detection using real-life trial data. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 59–66, 2015.

[10] Verónica Pérez-Rosas, Mohamed Abouelenien, Rada Mihalcea, Yao Xiao, CJ Linton, and Mihai Burzo. Verbal and nonverbal clues for real-life deception detection. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2336–2346, 2015.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[12] Zhe Wu, Bharat Singh, Larry Davis, and V Subrahmanian. Deception detection in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.