On the Use of Deep Recurrent Neural Networks for Detecting Audio Spoofing Attacks

S. Scardapane, L. Stoffl, F. Röhrbein and A. Uncini
Overview

1. Introduction

2. Dataset

3. Neural network architecture

4. Preliminary experimental results

5. Conclusions
Voice biometrics

Knowledge-based authentication
- 49% of users say that authentication is time-consuming

Voice Biometrics authentication
- 80% faster authentication in 5 seconds
- $15M average saving over a 3 year period
- 90% prefer Voice Biometrics over the status quo
- 85% of mobile users are frustrated with existing authentication

Figure: [http://www.sabio.co.uk/what-sabio-do/ivr-and-apps/voice-biometrics](http://www.sabio.co.uk/what-sabio-do/ivr-and-apps/voice-biometrics)
Wavenet

In 2016, Google showed how to obtain human-like voices using deep networks trained on the raw waveform:

![Wavenet diagram](image)

**Figure**: van den Oord, A., et al., 2016. *Wavenet: A generative model for raw audio*. CoRR abs/1609.03499.
Lyrebird

*New AI Tech can Mimic any Voice*
[Scientific American]

*Lyrebird claims it can recreate any voice using one minute of sample audio*
[The Verge]

*This audio clip of a robot as Trump may prelude a future of fake human voices*
[The Washington Post]

*This robot speech simulator can imitate anyone’s voice*
[The Telegraph]
Summary of the presentation

Objective

Investigating the use of deep neural networks for protecting against audio spoofing attacks in voice biometrics systems.

This talk

Preliminary results on a large dataset with a deep recurrent network show promising results, even in the absence of any post-processing of the output.
Overview

1. Introduction
2. Dataset
3. Neural network architecture
4. Preliminary experimental results
5. Conclusions
ASVspoof 2015 challenge

- We used the dataset released for the ASVspoof 2015 challenge.
- The focus was on voice conversion and speech synthesis attacks.
- There are 10 attacks in total, only 5 of which are known in the training phase.
- In total, there are 16651 genuine segments, and 246500 spoofed segments, divided into training, development, and test.
## Attacks for the challenge

<table>
<thead>
<tr>
<th>Type</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known</td>
<td>S1</td>
<td>Simple frame selection algorithm</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>Simple voice conversion algorithm(works with the first MFCC value)</td>
</tr>
<tr>
<td></td>
<td>S3</td>
<td>Speech synthesis algorithm based on HMMs and speaker adaptation</td>
</tr>
<tr>
<td></td>
<td>S4</td>
<td>Same as S3, using more data for the adaptation process</td>
</tr>
<tr>
<td></td>
<td>S5</td>
<td>Voice conversion algorithm based on the festvox project</td>
</tr>
<tr>
<td>Unknown</td>
<td>S6</td>
<td>Voice conversion based on GMM and ML parameter adaptation</td>
</tr>
<tr>
<td></td>
<td>S7</td>
<td>Similar to S6, using a different feature representation</td>
</tr>
<tr>
<td></td>
<td>S8</td>
<td>VC based on tensor decomposition with a Japanese dataset</td>
</tr>
<tr>
<td></td>
<td>S9</td>
<td>VC based on a kernel partial least-squares algorithm</td>
</tr>
<tr>
<td></td>
<td>S10</td>
<td>Speech synthesis using the Mary text-to-speech system</td>
</tr>
</tbody>
</table>
Evaluation of the system

The algorithms are evaluated by presenting them with the speech signals in the test set in a random order, and providing a score on whether the segments are genuine.

We define a false alarm probability given an alarm threshold $\theta$ as:

$$P_{fa}(\theta) = \frac{\# \{\text{spoofed trials with score} > \theta\}}{\# \{\text{total spoofed trials}\}}. \quad (1)$$

Similarly, we define a probability of missing a spoofed utterance:

$$P_{miss}(\theta) = \frac{\# \{\text{genuine trials with score} \leq \theta\}}{\# \{\text{total genuine trials}\}}. \quad (2)$$

The error measure is chosen as the equal error rate (EER), which is defined by choosing a value $\theta^*$ for the threshold such that $P_{fa}(\theta^*) = P_{miss}(\theta^*) = \text{EER}$. 
Overview

1. Introduction
2. Dataset
3. Neural network architecture
4. Preliminary experimental results
5. Conclusions
MFCC coefficient extraction

**Figure**: MFCC coefficients and Log-filterbank are extracted after segmenting the audio segment in 100 ms frames with no overlap.
Deep neural network

Figure: Architecture of the deep recurrent network used in the experiments.
Overview

1. Introduction
2. Dataset
3. Neural network architecture
4. Preliminary experimental results
5. Conclusions
Experiment setup

- We minimize the mean-squared error on the train set, weighted by an $\ell_2$ norm regularization.
- Gradients are computed via a truncated BPTT, where the output on an audio segment is defined by averaging the final 25 outputs of the network.
- Updates are performed on mini-batches of 500 utterances with the Adam optimization algorithm.
- Optimization is performed for a maximum of 150 epochs, and the development set is used to select an optimal regularization term.
- We randomly dropout neurons in the feedforward layers with probability 10% during training.
- All results are averaged over 15 different initializations of the weights.
## Results

<table>
<thead>
<tr>
<th>Feature set</th>
<th>Topology</th>
<th>Equal Error Rate (EER) [%]</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Known</td>
<td>Unknown</td>
</tr>
<tr>
<td>MFCC</td>
<td>3x Dense + 3x LSTM</td>
<td>(2.2, 3.4, 0.0, 0.2, 3.5)</td>
<td>3.9, 2.4</td>
<td>0.0, 2.8</td>
</tr>
<tr>
<td>LF</td>
<td>3x LSTM</td>
<td>(15.2, 15.3, 15.0, 15.0, 15.3)</td>
<td>(15.3, 15.3, 15.0, 15.4, 37.7)</td>
<td></td>
</tr>
<tr>
<td>MFCC + LF</td>
<td>3x LSTM</td>
<td>(6.5, 9.0, 4.4, 4.2, 10.1)</td>
<td>(10.3, 7.5, 2.8, 8.4, 38.1)</td>
<td></td>
</tr>
<tr>
<td>MFCC + LF</td>
<td>3x Dense + 3x LSTM</td>
<td>(0.3, 0.7, 0.4, 0.4, 0.9)</td>
<td>(0.9, 0.6, 0.5, 0.7, 96.0)</td>
<td></td>
</tr>
</tbody>
</table>
## Comparisons

<table>
<thead>
<tr>
<th>System</th>
<th>Equal Error Rate (EER) [%]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Known</td>
<td>Unknown</td>
</tr>
<tr>
<td>A</td>
<td>0.408</td>
<td>2.013</td>
</tr>
<tr>
<td>B</td>
<td>0.008</td>
<td>3.922</td>
</tr>
<tr>
<td>C</td>
<td>0.058</td>
<td>4.998</td>
</tr>
<tr>
<td>D</td>
<td><strong>0.003</strong></td>
<td>5.231</td>
</tr>
<tr>
<td>E</td>
<td>0.041</td>
<td>5.347</td>
</tr>
<tr>
<td><strong>Proposed</strong></td>
<td>0.040</td>
<td>3.960</td>
</tr>
</tbody>
</table>
Overview

1. Introduction
2. Dataset
3. Neural network architecture
4. Preliminary experimental results
5. Conclusions
Conclusions

1. Deep RNN architectures can reach state-of-the-art results in the anti-spoofing task, even with a small number of layers and a minimal amount of fine-tuning.

2. We can think of training the model on the raw audio data to see what features the network can learn in order to discriminate among spoofing and non-spoofing attacks.

3. We envision their use on more complex spoofing attacks, or a combination of several basic attacks.

4. Deep RNNs can also be integrated in existing voice recognition, language identification, and speaker classification systems, resulting in extremely smart multitask models all exploiting a common infrastructure.