End-to-end speaker spoofing detection

Heinrich Dinkel, Nanxin Chen, Yanmin Qian, Kai Yu
Shanghai Jiao Tong University
Outline

- Intro
  - Speaker verification
  - Speaker spoofing attacks

- Spoofing
  - Countermeasures
  - Corpus
  - Motivation

- Deep Learning
  - CLDNN
  - Results
Intro
Speaker verification

• **Purpose:** Secure assets over voice “voice fingerprint”

• **Structure:**
  ○ Train [Background Model]
  ○ Enrol [Few utterances]
  ○ Eval [Utterance → Score → Decision]

• **Metric:**
  ○ False Acceptance Rate (FAR)
  ○ False Rejection Rate (FRR)
  ○ Equal Error Rate (EER),
    Half Total Error Rate (HTER)
Spoof detection - Attacks

- Impersonation
- Replay
- Voice conversion (VC)
- Text-to-speech (TTS)
Spoofing detection - Example system

Evaluation Utterance + Claimed ID

Score >= Threshold $\theta_{sp}$ ?

Counter measure

Claim = True? ASV

Score >= Threshold $\theta_{asv}$ ?

Reject Access

Reject

Heinrich Dinkel, Shanghai Jiao Tong University, End-to-end speaker spoofing detection
Corpus: BTAS 2016

- Impersonation
  - Focus: Replay Attacks (VC, TTS also)
  - Different “Quality” Attacks (Microphone, Speaker)
  - Evaluation has unseen replay (Focus)
  - HTER as measure

<table>
<thead>
<tr>
<th>Type</th>
<th>Train</th>
<th>Dev</th>
<th>Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genuine</td>
<td>4973</td>
<td>4995</td>
<td>5576</td>
</tr>
<tr>
<td>Attacks</td>
<td>38580</td>
<td>38580</td>
<td>44920</td>
</tr>
<tr>
<td>TTS</td>
<td>2.5%</td>
<td>2.5%</td>
<td>2.5%</td>
</tr>
<tr>
<td>VC</td>
<td>90%</td>
<td>90%</td>
<td>87%</td>
</tr>
<tr>
<td>Replay (K)</td>
<td>7.5%</td>
<td>7.5%</td>
<td>7%</td>
</tr>
<tr>
<td>Replay (U)</td>
<td>-</td>
<td>-</td>
<td>3.5%</td>
</tr>
</tbody>
</table>
BTAS2016 - Evaluation

- Uses HTER, computed from the development set threshold:

\[
\theta_{\text{dev}} = \arg \min_{\theta} \frac{\text{FAR}_{\text{dev}}(\theta) + \text{FRR}_{\text{dev}}(\theta)}{2}
\]

\[
\text{HTER}_{\text{eval}} = \frac{\text{FAR}_{\text{eval}}(\theta_{\text{dev}}) + \text{FRR}_{\text{eval}}(\theta_{\text{dev}})}{2}
\]
Countermeasures

- Standard: Feature + Classifier
- Cepstral features
  - Mel cepstrum
  - Perceptual Linear Predictive
  - Constant Q
  - Gammatone Frequency
- Gaussian mixture model
- Identity Vector (I-Vector)
- Deep feature approach
Countermeasures - Deep features

- Extension of classic feature + classifier
- Input: Feature
  Output: Class Label
  Purpose: Extract spoofing vector (s-vector)
- Final classifier: GMM, LDA, SVM
Corpus: Countermeasures and Baseline of BTAS2016

- Spoof-aware features
- Features > Classifiers
- Aim: Outperform 1st

<table>
<thead>
<tr>
<th>Position</th>
<th>Feature</th>
<th>Classifier</th>
<th>HTER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3rd</td>
<td>PLP-39</td>
<td>BLSTM-DNN</td>
<td>2.20</td>
</tr>
<tr>
<td>2nd</td>
<td>MCEP</td>
<td>LDA</td>
<td>2.04</td>
</tr>
<tr>
<td>1st</td>
<td>MFCC+i-MFCC</td>
<td>GMM</td>
<td>1.26</td>
</tr>
</tbody>
</table>
Motivation and Model proposal
Motivation

- Features > Classifier
- Two “independent” tasks: feature + classifier
- Non-task optimized feature (trial + error)
- Classifier parameter (trial + error)

Why not both?
Convolutional Long Short Term Neural Networks (CLDNN)

- Proposed by Google [Learning the Speech Front-end With Raw Waveform CLDNNs]
- Front-end feature extractor (CNN)
- Sequence-classification (LSTM)
- Improved Accuracy (DNN)

All in one model
Model - Time frequency CNN

Time Pooling

- Time convolution
- Pooling
- \([1 \times F \times C] \rightarrow [P_T \times F \times C] \rightarrow [P_T \times 1 \times 1]\)

Frequency Conv

- \([P_f \times P_T \times 1]\)
- \([P'_f \times P'_T \times 1]\)

Similar to fast fourier transform
- Extracts feature
- Enhances invariance
- Operations only over one dimension

Heinrich Dinkel, Shanghai Jiao Tong University, End-to-end speaker spoofing detection
Model - LSTM

- Each input → Output
- We only have one label / each utterance
- Many-to-one mapping
- Last timestep is used as representation
Model - Classifier

- Standard neural network (512 hidden neurons)
- Maps LSTM prediction to error
- Enhanced by a 50% dropout layer
Model description - Overview

$S = 25$
$F = 560$
$C = 1$

$p_T' = 39$
$p_f' = 128$

2 $\square$ 128 LSTM
1 $\square$ 512 DNN

Heinrich Dinkel, Shanghai Jiao Tong University, End-to-end speaker spoofing detection
Experiments
Experiment - Feature details

- Samplerate 16kHz, Converted 32bit data (replay) → 16 bit (others)
- Input is 35ms window frame (560)
- Window shift by 12.5ms (200)
- Sequence length of 25
- 50% Dropout in Classifier
- Adadelta optimization (no learning rate)
- 3 Iterations
- 5 Output neurons (Genuine + 4 Spoof) [merged HQ+LQ]
## Results

<table>
<thead>
<tr>
<th>Attack</th>
<th>MFCC+i-MFCC+GMM</th>
<th>CLDNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>1.26%</td>
<td>0.82%</td>
</tr>
<tr>
<td>TSS</td>
<td>0.68%</td>
<td>0.51%</td>
</tr>
<tr>
<td>VC</td>
<td>0.75%</td>
<td>0.41%</td>
</tr>
<tr>
<td>Replay (Known)</td>
<td>1.01%</td>
<td>0.77%</td>
</tr>
<tr>
<td>Replay (Unknown)</td>
<td>14.78%</td>
<td>11.24%</td>
</tr>
</tbody>
</table>

All results in HTER%
Summary

- Neural network + raw wave does work (First)
- End to end processing simplifies pipeline
- Capable of generalization (unseen attacks)
- Can also be used as feature extractor (future experiments)
Thanks!

Questions?

heinrich.dinkel@gmail.com