End-to-end speaker spoofing detection

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

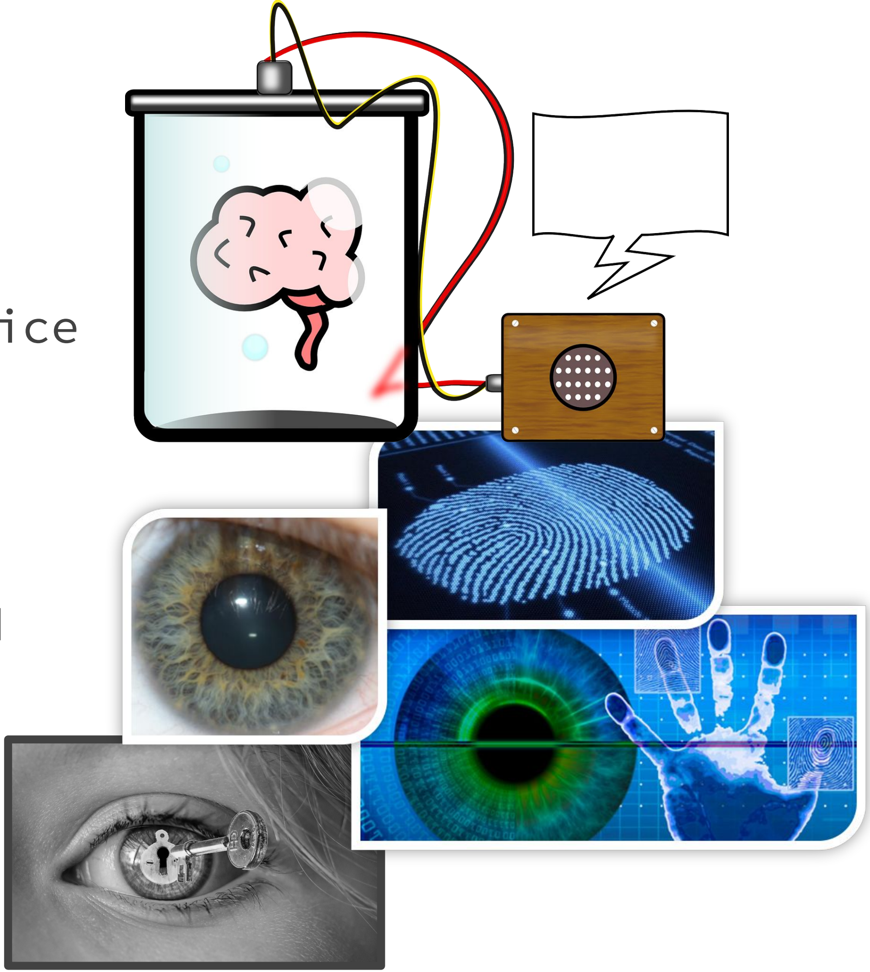
Outline

- Intro
  - Speaker verification
  - Speaker spoofing attacks
- Spoofing
  - Countermeasures
  - Corpus
  - Motivation
- Deep Learning
  - CLDNN
  - Results

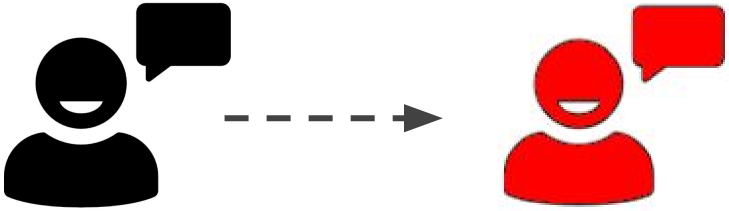
Intro

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)

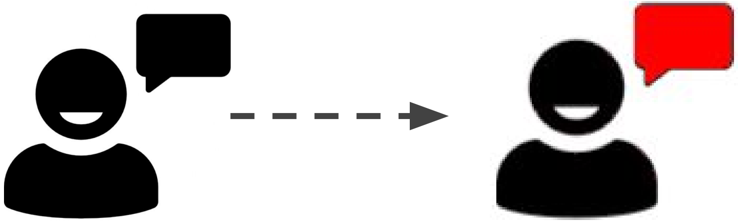


Spoof detection - Attacks

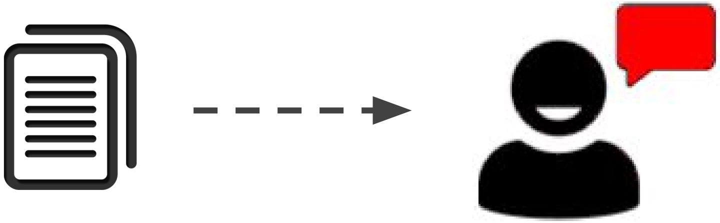


Impersonation

Replay

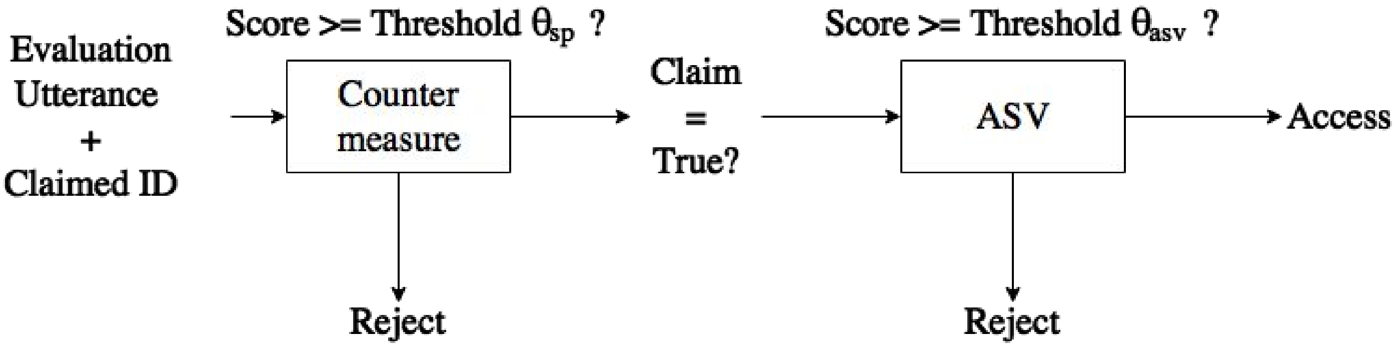


Voice conversion (VC)



Text-to-speech (TTS)

Spoofing detection - Example system


Score >= Threshold θ_{sp} ?
Evaluation Utterance + Claimed ID
Counter measure
Reject
Claim = True?
Score >= Threshold θ_{asv} ?
ASV
Reject
Access

Corpus: BTAS 2016

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

Type	Train	Dev	Eval
Genuine	4973	4995	5576
Attacks	38580	38580	44920
TTS	2.5%	2.5%	2.5%
VC	90%	90%	87%
Replay (K)	7.5%	7.5%	7%
Replay (U)	-	-	3.5%

BTAS2016 - Evaluation

- Uses HTER, computed from the development set threshold:

$$\theta_{dev} = \arg\min_{\theta} \frac{\mathrm{FAR}_{\mathrm{dev}}(\theta) + \mathrm{FRR}_{\mathrm{dev}}(\theta)}{2}$$

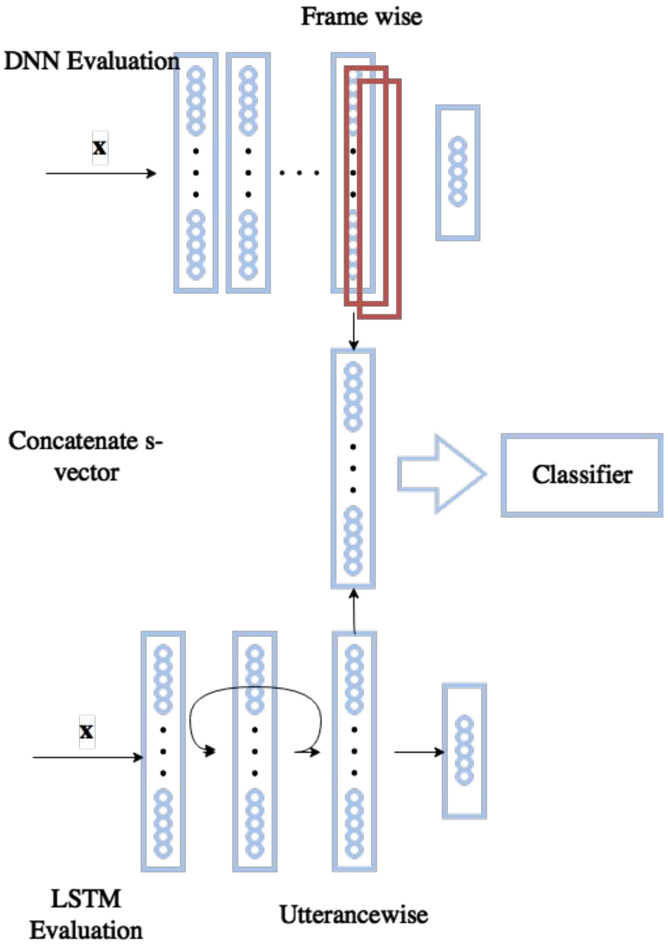
$$\mathrm{HTER}_{\mathrm{eval}} = \frac{\mathrm{FAR}_{\mathrm{eval}}(\theta_{\mathrm{dev}}) + \mathrm{FRR}_{\mathrm{eval}}(\theta_{\mathrm{dev}})}{2}$$

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

Position	Feature	Classifier	HTER (%)
3rd	PLP-39	BLSTM-DNN	2.20
2nd	MCEP	LDA	2.04
1st	MFCC+i-MFCC	GMM	1.26

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?

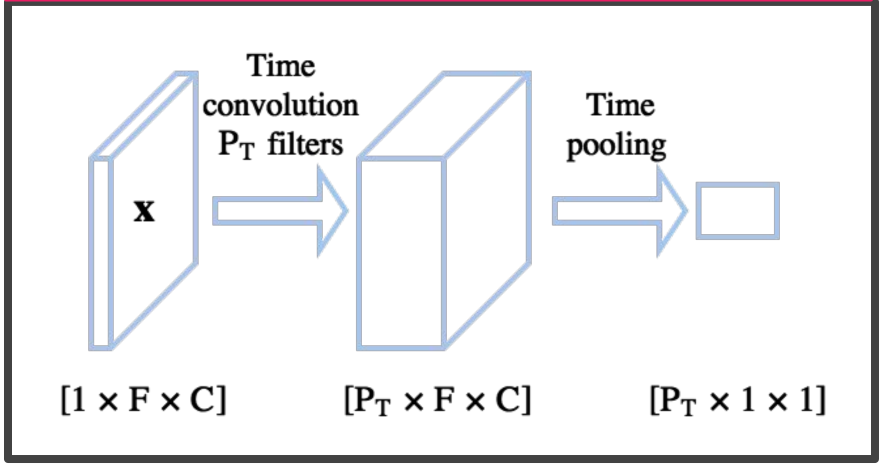
Convolutional Long Short Term Neural Networks (CLDNN)

- 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

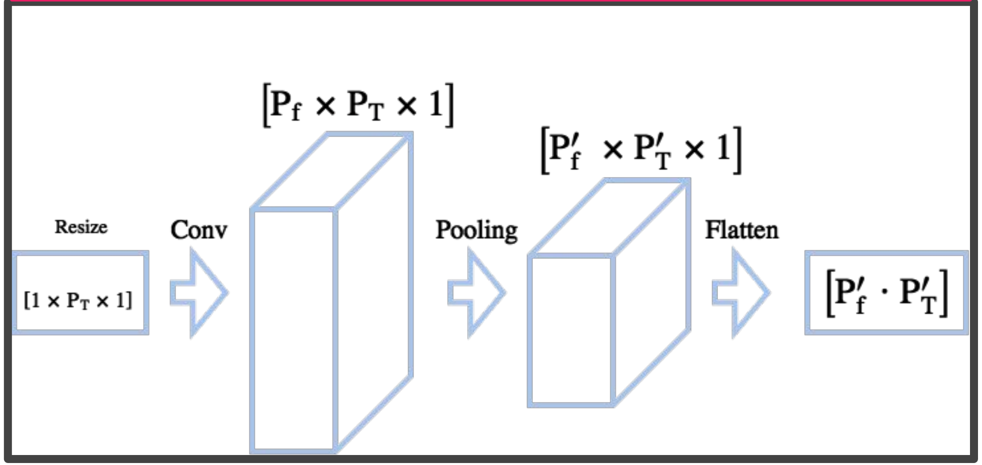
Time Pooling


Time convolution P_T filters
Time pooling
x
$[1 \times F \times C]$
$[P_T \times F \times C]$
$[P_T \times 1 \times 1]$

Similar to fast fourier transform

Extracts feature

Frequency Conv

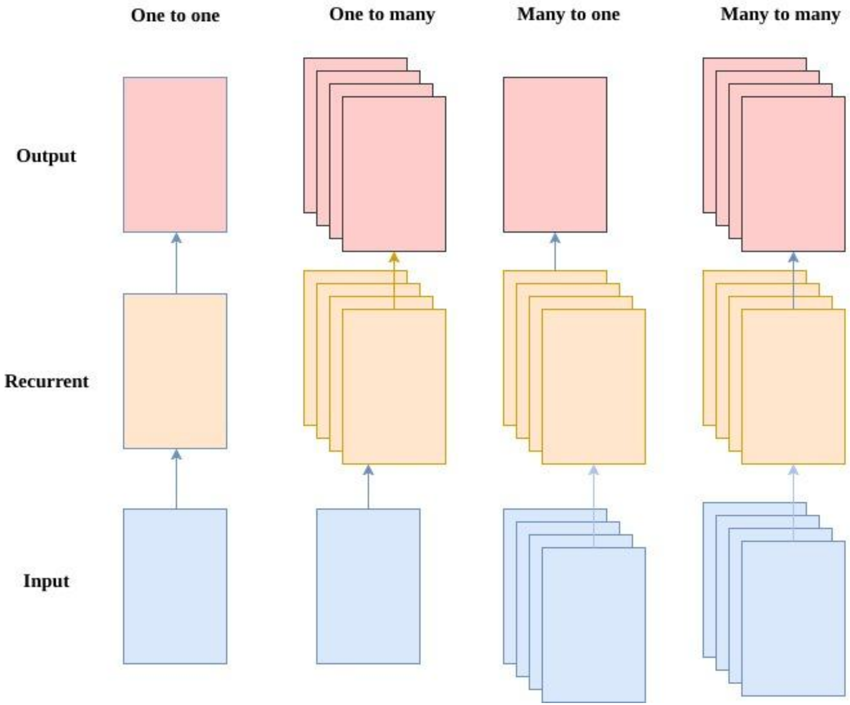

Resize
$[1 \times P_T \times 1]$
Conv
$[P_f \times P_T \times 1]$
Pooling
$[P'_f \times P'_T \times 1]$
Flatten
$[P'_f \cdot P'_T]$

Enhances invariance

Operations only over one dimension

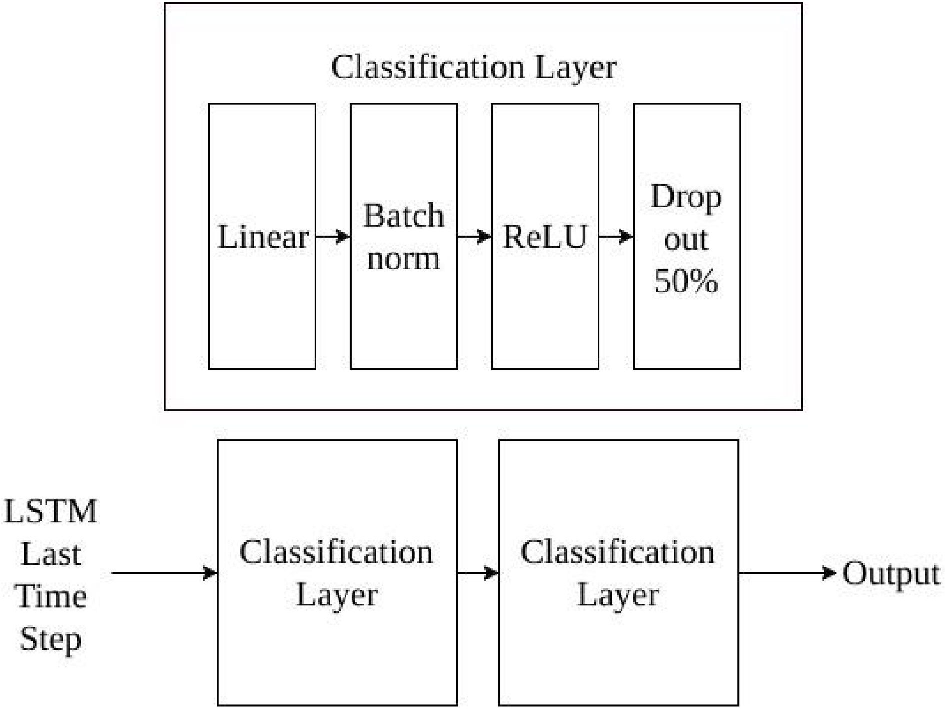
Model - LSTM

- 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

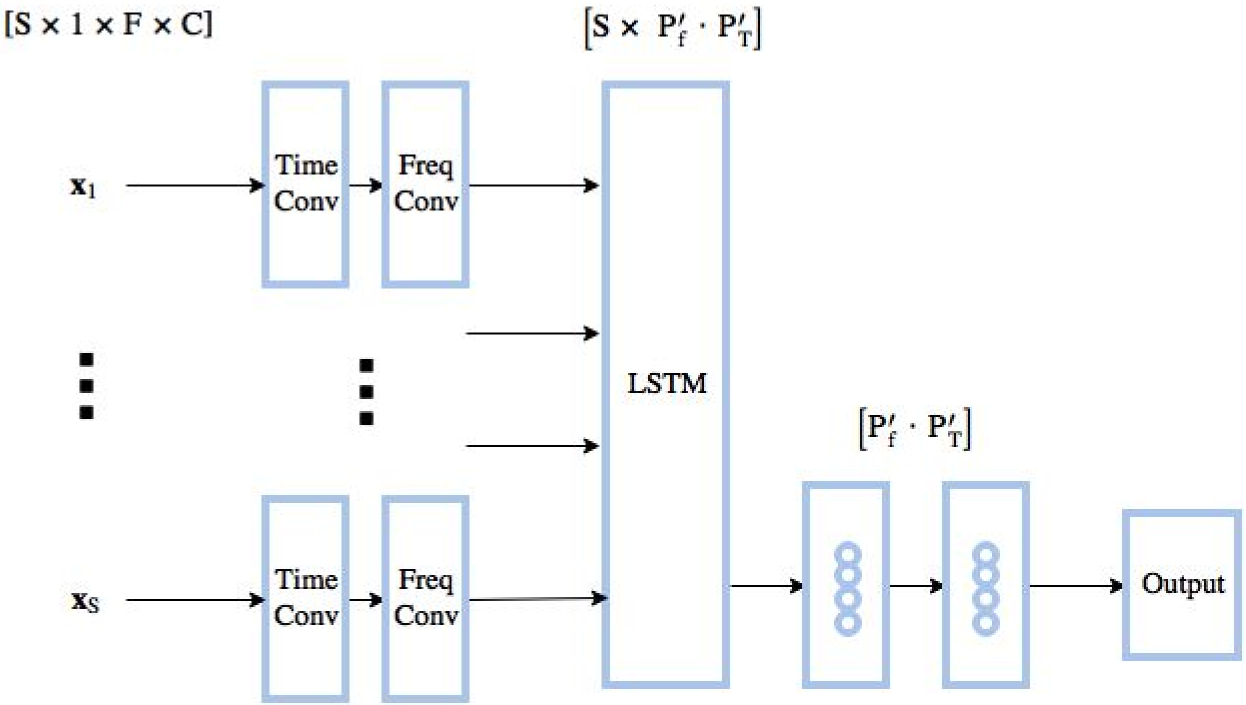
C = 1

$P_T' = 39$

$P_f' = 128$

2 □ 128 LSTM

1 □ 512 DNN



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

Attack	MFCC+i-MFCC+GMM	CLDNN
All	1.26%	**0.82%**
TSS	0.68%	0.51%
VC	0.75%	0.41%
Replay (Known)	1.01%	0.77%
Replay (Unknown)	14.78%	11.24%

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