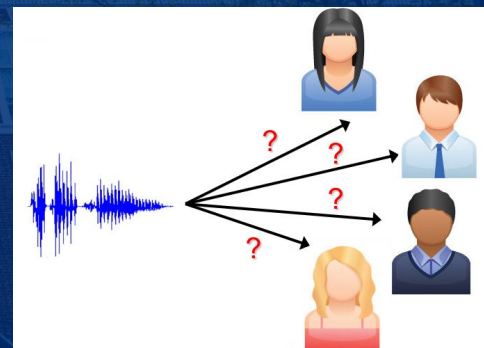




JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Speaker Verification using Machine and Deep Learning Approaches on the VoxCeleb Dataset

Yixiong Chen, Zuojun Zhou, Susan Wang, Yuhao Zheng
Dec. 6th, 2024



Introduction

Speak Verification:

- recognize who is speaking from a clip of audio recordings accurately.

Importance:

- Device unlocking, Touchless control, create personalization, etc.
- huge real-world usage with a focus on security and usability.



Project Goal

- Build >1 models that can accurately recognize who is speaking from audio recordings
- Compare the performance of traditional Machine Learning and Deep Learning approaches

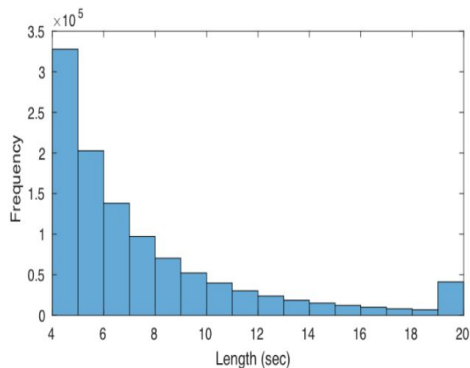
Dataset Overview: VoxCeleb



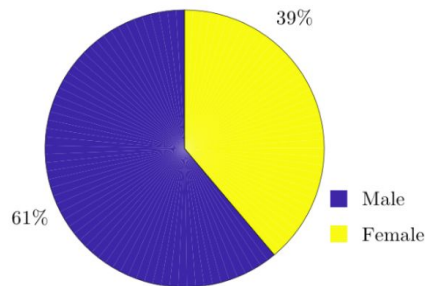
Dataset Overview

Key Features:

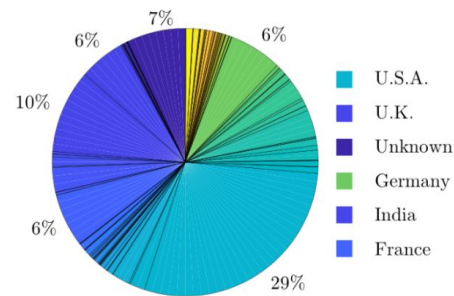
- Diversity and scale
- Real-world noises and variability
- Rich features: opportunity for preprocessing



Utterance Lengths



Gender Distribution



Nationality Distribution

Data Preprocessing

- **Normalization**
- resample to 16 kHz
 - maintain uniform time resolution
- trimmed to remove silence and background noise
 - focus on speech segments
- 100-120 audio clips per speaker
- every audio clip is 4-6 seconds

Feature Extraction

For traditional ML models:

- **Pitch:** Mean and standard deviation of fundamental frequency (f_0)
- **Mel-Frequency Cepstral Coefficients (MFCCs):** 13 Mel-Frequency Cepstral coefficients and their deltas.
- **Spectral Features:** Centroid, contrast, and chroma.
- **Energy, Zero-Crossing Rate (ZCR):** Capture signal dynamics.

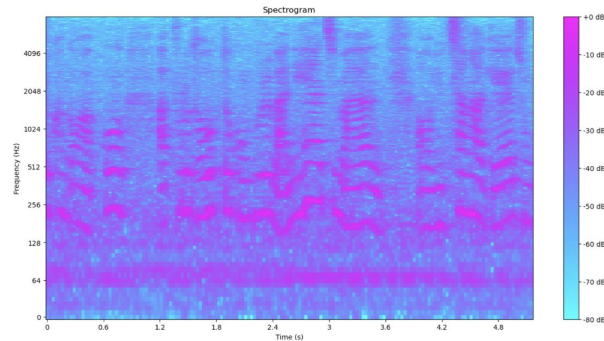
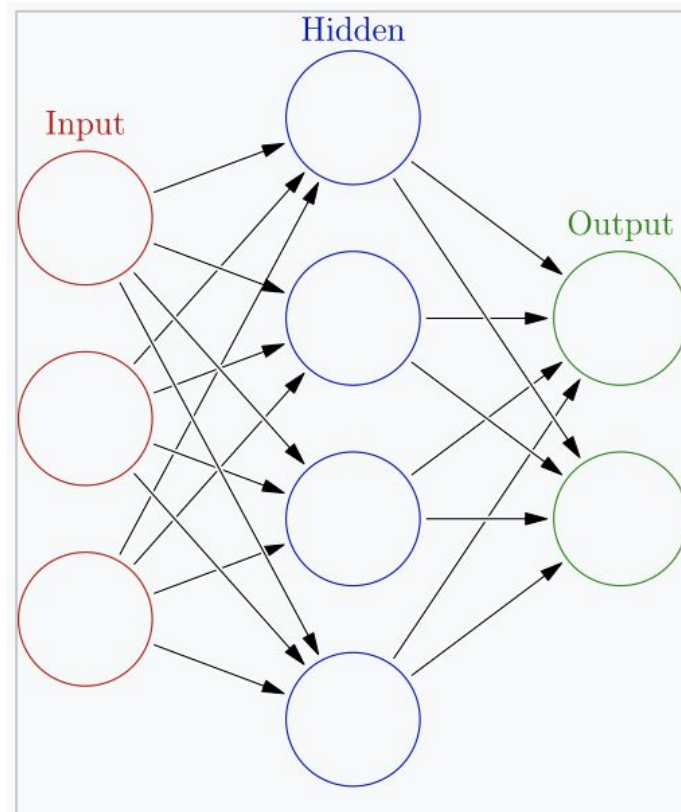


Figure 1: A spectrogram representation of an audio segment

Feature Extraction

For DL models:

- Direct use of raw audio or log-Mel spectrograms.
- hand-crafted features (YAMNet)



Methodology: ML Models

1. Naive Bayes

2. Logistic Regression

3. Support Vector Machines (SVM):

- Kernels: Linear, RBF.
- Tuned Parameters using grid search: C (regularization), γ (kernel coefficient).

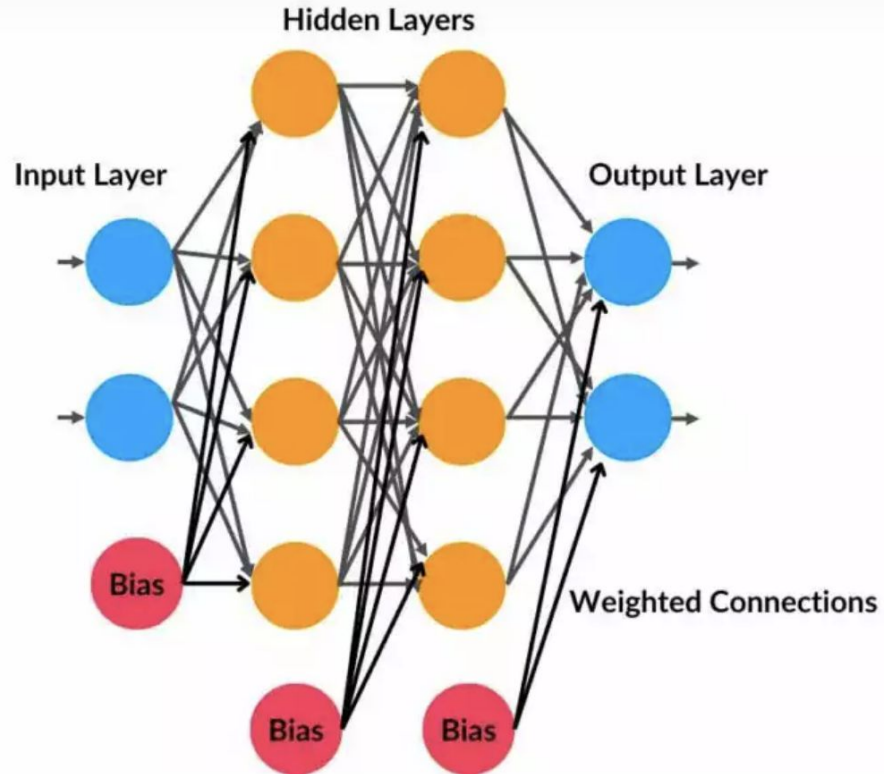
4. Random Forest: Decision trees optimized for depth and estimators.

5. XGBoost: Advanced tree-based model

Methodology: DL Models

Three Multilayer Perceptron (MLP) Architectures:

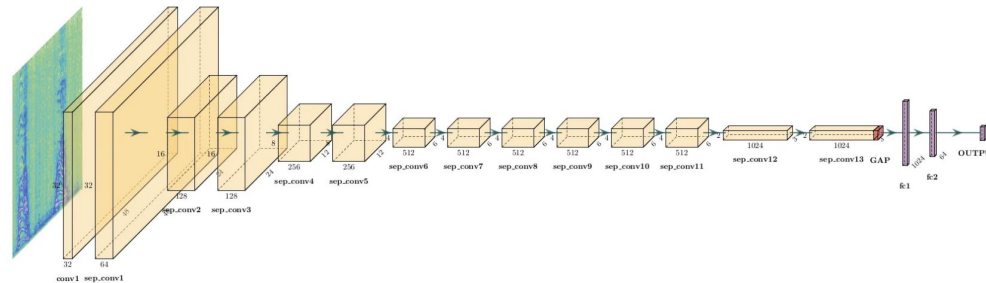
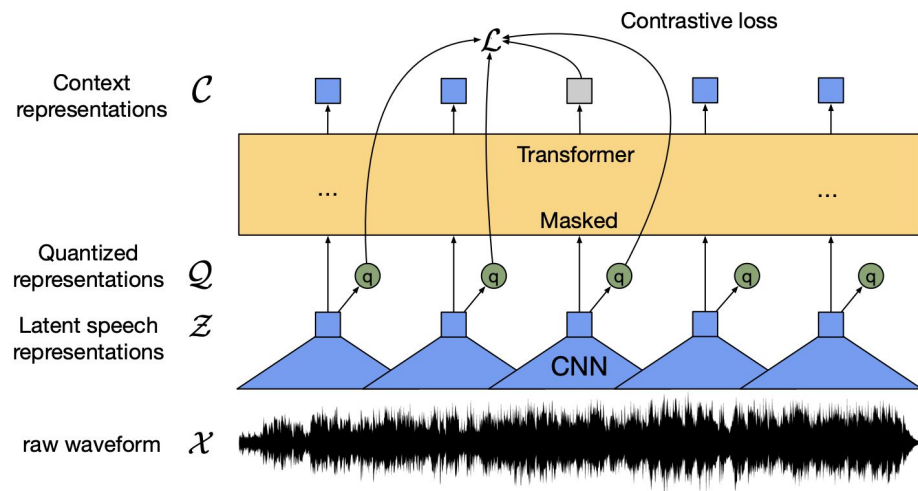
- SmallNet (2 layers), MediumNet (3 layers), LargeNet (4 layers).
- trained from scratch using extracted embeddings.



Methodology: DL Models

Pre-trained Models (on 200-speaker subset):

- Wav2Vec: Fine-tuned for speaker classification.
- YAMNet: Adapted from sound event detection to speaker tasks.



Evaluation Metrics

Accuracy:

- % of correctly classified samples.
- Primary metric for comparison.

Training Time:

- Qualitative

We balance clarity in comparisons and real-world feasibility.

Experiment Setup

Dataset: VoxCeleb (200-speaker subset used for consistency).

Data split: 80% training, 20% testing (no speaker overlap).

Inputs:

- **ML:** Hand-crafted features (e.g., MFCCs, ZCR).
- **DL:** Raw audio or spectrograms, plus hand-crafted features (YAMNet)

Baseline vs. fine-tuned models tested.

Feature Selection

=== Trying threshold: 0.01 ===

Number of features selected: 39

Selected features: [4, 36, 14, 6, 10, 12, 17, 22, 5, 1, 2, 15, 3, 0, 23, 7, 11, 9, 8, 16, 38, 21, 30, 29, 18, 20, 27, 13, 31, 19, 28, 33, 34, 24, 25, 26, 32, 35, 37]

Cross-validated accuracy: 0.4757 ± 0.0640

=== Trying threshold: 0.015 ===

Number of features selected: 29

Selected features: [4, 36, 14, 6, 10, 12, 17, 22, 5, 1, 2, 15, 3, 0, 23, 7, 11, 9, 8, 16, 38, 21, 30, 29, 18, 20, 27, 13, 31]

Cross-validated accuracy: 0.4741 ± 0.0662

=== Trying threshold: 0.02 ===

Number of features selected: 22

Selected features: [4, 36, 14, 6, 10, 12, 17, 22, 5, 1, 2, 15, 3, 0, 23, 7, 11, 9, 8, 16, 38, 21]

Cross-validated accuracy: 0.4685 ± 0.0670

=== Trying threshold: 0.025 ===

Number of features selected: 16

Selected features: [4, 36, 14, 6, 10, 12, 17, 22, 5, 1, 2, 15, 3, 0, 23, 7]

Cross-validated accuracy: 0.4365 ± 0.0611

Selected Features:

All features except

- Zero center rate
- energy

Machine Learning Results

Model	Validation Accuracy (%)
Naive Bayes	17.56
Logistic Regression	23.99
Support Vector Machine (SVM)	28.71
Random Forest	25.47
XGboost	25.20

Table 1: Models Performance with traditional 3 audio features

Machine Learning Results

Model	Validation Accuracy (%)
Naive Bayes	24.15
Logistic Regression	36.41
Support Vector Machine (SVM)	37.61
Random Forest	33.83
XGboost	35.35

Table 3: Models Performance with newly added features

Machine Learning Results

Hyperparameter Optimization

- After extensive tuning,
optimized **Random Forest** achieved **33.83%** accuracy
- Key tuned parameters:
 - N estimators: 100
 - Max depth: 50
 - Min sample split: 2
 - Min sample leaf: 1

Machine Learning Results

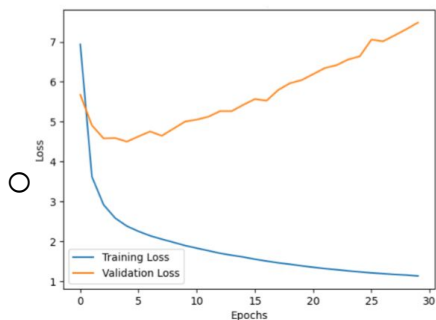
Hyperparameter Optimization

- After extensive tuning,
optimized **XGBoost** achieved **35.35%** accuracy
- Key tuned parameters:
 - Learning rate: 0.05
 - Max depth: 10
 - Estimators: 1000
 - Subsample: 0.4

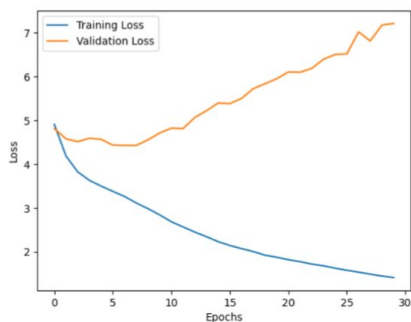
Deep Learning Results

MLP Models (predefined features)

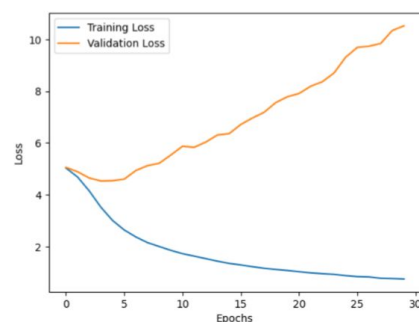
Model	Training Accuracy	Val. Accuracy	Training Speed
SmallNet	69.94%	21.99%	1.08s/epoch
MediumNet	63.83%	20.48%	1.29s/epoch
LargeNet	78.49%	21.17%	1.57s/epoch



(a) SmallNet



(b) MediumNet

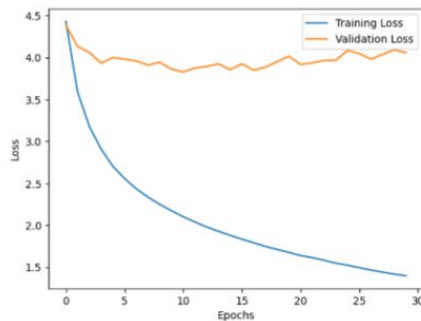


(c) LargeNet

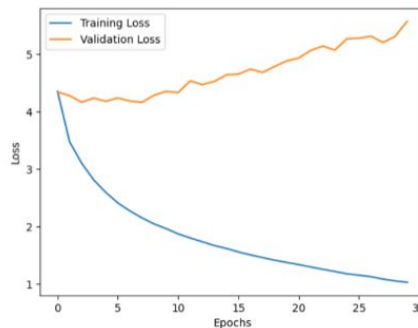
Deep Learning Results

YAMNet:

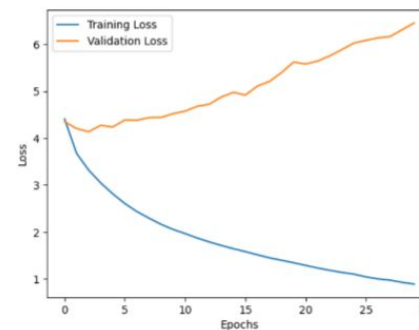
Model	Training Accuracy	Val. Accuracy	Training Speed
SmallNet	64.45%	23.91%	1.21s/epoch
MediumNet	70.88%	21.40%	1.36s/epoch
LargeNet	74.09%	20.08%	1.49s/epoch



(a) SmallNet



(b) MediumNet



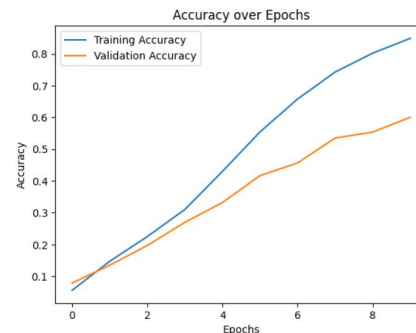
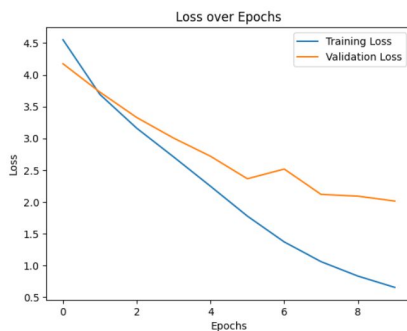
(c) LargeNet

Deep Learning Results

Fine-Tuned Models

- **Wav2Vec:**

- Training: 84.92%; Validation: 60.03%. (Took about 7 hours (3.8 iterations per second) to train the model for 10 epochs)



- **HuBERT:** Not fine-tuned; high computational cost.

- One iteration is ≈ 17 seconds for a batch of two audio clips, equivalent to 203 hours for an epoch.

Overall Performance Comparison

Deep Learning Outperforms Traditional Machine Learning

- Wav2Vec Model: Validation Accuracy: 60.03%
- Best ML Model (SVM): Validation Accuracy: 37.61%

Performance Gap:

- DL models effectively capture complex speaker-specific features.
- Traditional ML relies on manually designed features (MFCCs, ZCR)

Computational Considerations

Traditional ML Models:

- Hardware: Trained on Apple M2 Pro CPU
- Pros: Quick training iterations, low resource requirements

Deep Learning Models:

- Hardware: Trained on AMD Ryzen 5900X CPU & NVIDIA RTX 3060 GPU
- Cons: Higher computational demands, longer training times

Conclusion, Future Work

DL models (Wav2Vec) outperform ML in speaker verification.

ML still works for lightweight applications.

Future Directions:

- Making deep learning methods more resource-efficient.
- Optimize training pipelines to reduce computational overhead without losing performance. (Pre-process and cache audio features)



Thank You!
Any Questions?

References

- [1] McFee, B., et al. "librosa: Audio and music signal analysis in python." Proceedings of the 14th Python in Science Conference. 2015.
- [2] Chen, T., and Guestrin, C. "XGBoost: A scalable tree boosting system." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016.
- [3] Schneider, S., Baevski, A., Collobert, R., Auli, M. (2019). "Wav2vec: Unsupervised pre-training for speech recognition." In Advances in Neural Information Processing Systems (pp. 8012-8022).
- [4] Hsu, W.-N., Heigold, G., Bollegala, D., Chen, J., Eisenschlos, J., Raffel, C., Auli, M. (2021). "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units." arXiv preprint arXiv:2106.07447.
- [5] Campbell, N., Klatt, D., Bergstra, J., Bergstra, S. (2020). "YAM-Net: A Pretrained Audio Event Classification Model and Dataset." arXivpreprint arXiv:2009.03029.
- [6] Praat Developers. "pyin: Pitch tracking." <https://librosa.org/doc/main/generated/librosa.pyin.html>
- [7] Loshchilov, I., Hutter, F. (2017). "Decoupled Weight Decay Regularization." arXiv preprint arXiv:1711.05101.
- [8] TensorFlow Hub Team. "TensorFlow Hub." <https://www.tensorflow.org/hub>.