Audio Deepfake Detection

George Nazos (MTN2519) Konstantinos Kaimakis (MTN2508)

A project for the MSc in Artificial Intelligence at the NCSR Demokritos - University of Piraeus

February 11, 2026

Outline

1. Introduction
2. Feature Extraction & Selection
3. Methodology
4. Results on FoR Test Set
5. Zero-Shot Evaluation
6. Discussion
7. Conclusion

Problem Statement

- Audio deepfakes generated by modern TTS and voice-cloning systems are increasingly realistic and difficult to detect by ear.
- **Goal:** Build a classifier using machine learning methods that distinguishes *real* human speech from *fake* (machine-generated) speech using handcrafted audio features.
- We train on the **Fake-or-Real (FoR)** dataset and evaluate zero-shot generalization on two unseen datasets.

Datasets

Dataset	Samples	Role	Source
FoR (train)	53.864	Training + Validation	Controlled TTS
FoR (test)	4.634	In-distribution test	Controlled TTS
In-the-Wild	31.526	Zero-shot evaluation	Various real-world
ElevenLabs	136	Zero-shot evaluation	Single TTS provider

- **In-the-Wild**: diverse sources, varied recording conditions.
- **ElevenLabs**: single synthetic voice provider using our personal recordings

Extracted Audio Features

For every audio clip we extract frame-level features using the librosa python package.

Feature Group	**Description**	**Coefficients**
MFCC	Mel-frequency cepstral coefficients	20
MFCC Δ	First-order temporal derivatives	20
MFCC Δ^2	Second-order temporal derivatives	20
Mel Spectrogram	Energy per mel-frequency band	128
Spectral	Centroid, bandwidth, flatness, rolloff	4
Other	RMSE, ZCR, pitch (YIN)	3

Each feature was computed as both **mean** and **standard deviation** across all frames ($\sim$394 total features).

Feature Selection via Logistic Regression

We trained a Logistic Regression model on three feature subsets and compared prediction scores on the FoR validation set:

Feature Set	**Features**	**F1**	**ROC AUC**
All (mean + std)	∼394	0.72	0.80
Without mel spectrogram	∼138	0.55	0.56
Mean only (selected)	**195**	**0.78**	**0.85**

Key findings:

- Removing mel spectrogram features collapsed performance ⇒ they are essential.
- Dropping std aggregations *improved* every metric ⇒ std added noise.
- Final set: **195 mean-only features** used for all models.

Models & Grid Search

Nine models were trained using `grid_search_joblib` with F_1 (macro) scoring:

Model	**Key Hyperparameters Searched**
Logistic Regression	C, penalty
Linear SVM	$C \in \{0.001 \dots 1000\}$
RBF SVM	C, γ
Polynomial SVM	C, γ, degree, coef0
Sigmoid SVM	C, γ, coef0
Decision Tree	max_depth, min_samples_split, min_samples_leaf, criterion, ccp_alpha
Random Forest	n_estimators, max_depth, max_samples (split - leaf), max_features
XGBoost	max_depth, learning_rate, subsampling, colsample_bytree, scale_pos_weight ...
XGBoost (no mel)	Same as XGBoost, 67 features

- Best model retrained on **train + validation**, evaluated on **test**.
- Class weighting was applied to the fake class to prioritize its detection.
- Linear models use `StandardScaler`;

FoR Test Set – All Models

Model	Acc.	Prec.	Recall	F1	AUC
Logistic Reg.	0.728	0.787	0.722	0.71	0.902
Linear SVM	0.705	0.774	0.699	0.681	0.890
RBF SVM	0.761	0.761	0.76	0.761	0.832
Poly SVM	0.761	0.80	0.767	0.755	0.848
Sigmoid SVM	0.727	0.766	0.722	0.714	0.835
Decision Tree	**0.954**	**0.956**	**0.954**	**0.954**	**0.959**
Random Forest	0.766	0.786	0.763	0.76	0.873
XGBoost	0.921	0.921	0.921	0.921	0.973
XGBoost (no mel)	0.739	0.744	0.74	0.738	0.824

- **Decision Tree** achieves the best overall FoR test performance.
- Linear models (LR, Linear SVM) are the most consistent.
- Decision Tree and XGBoost show signs of overfitting.

Zero-Shot: In-the-Wild

Models evaluated on the In-the-Wild Dataset:

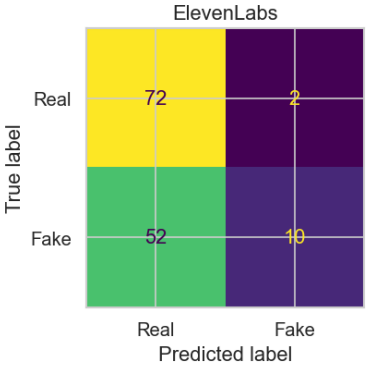
Model	Acc.	Prec.	Recall	F1	AUC
Logistic Reg.	**0.716**	**0.706**	**0.718**	**0.707**	**0.777**
Linear SVM	**0.720**	**0.707**	**0.718**	**0.709**	**0.781**
RBF SVM	0.706	0.696	0.708	0.697	0.763
Poly SVM	0.674	0.649	0.613	0.614	0.690
Sigmoid SVM	0.66	0.676	0.686	0.658	0.704
Decision Tree	0.59	0.549	0.545	0.544	0.521
Random Forest	0.64	0.6	0.565	0.554	0.65
XGBoost	0.665	0.695	0.564	0.526	0.728
XGBoost (no mel)	**0.741**	**0.731**	**0.698**	**0.705**	**0.785**

Zero-Shot: ElevenLabs

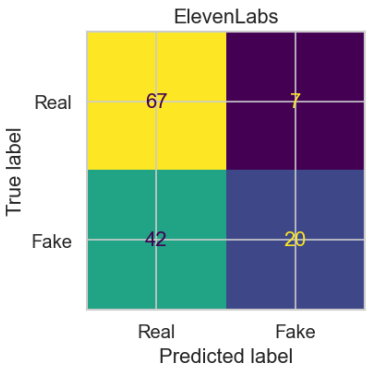
Models evaluated on the ElevenLabs Dataset:

Model	Acc.	Prec.	Recall	F1	AUC
Logistic Reg.	0.603	0.707	0.567	0.499	0.654
Linear SVM	0.610	0.716	0.575	0.512	0.671
RBF SVM	0.573	0.697	0.533	0.431	0.727
Poly SVM	0.544	0.272	0.5	0.352	0.487
Sigmoid SVM	**0.639**	**0.677**	**0.614**	**0.591**	**0.612**
Decision Tree	**0.669**	**0.785**	**0.638**	**0.604**	**0.773**
Random Forest	0.544	0.272	0.5	0.352	**0.827**
XGBoost	0.544	0.272	0.5	0.352	**0.664**
XGBoost (no mel)	0.544	0.272	0.5	0.352	**0.672**

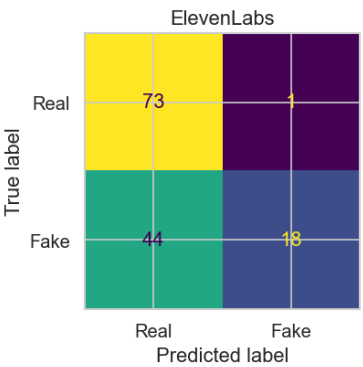
Confusion Matrix Highlights



Logistic Regression

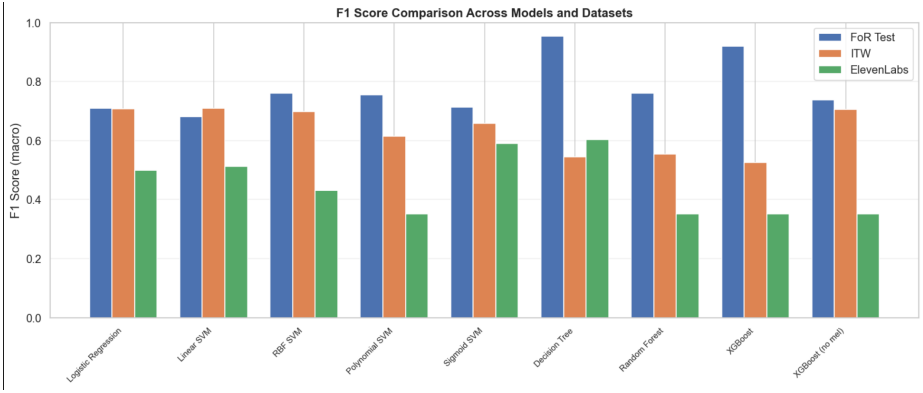


Sigmoid SVM



Decision Tree

F1 Score Comparison



Impact of Mel Spectrogram Features

XGBoost (195 features, with mel):

- FoR test F1 = 0.526
- ROC AUC = 0.728
- High mel correlation $\Rightarrow$ overfitting

XGBoost (67 features, no mel):

- FoR test F1 = **0.706**
- ROC AUC = **0.786**
- Better generalization

- Removing mel features **improved** XGBoost performance by +0.18 F1.
- For Logistic Regression, mel features were **essential** (F1 dropped from 0.78 to 0.55 without them).
- **Conclusion:** The effect of mel features is model-dependent — linear models benefit, tree-based models overfit on them.

Specificity vs. Generality Trade-off

- **Decision Tree** (deep, specific rules):
  - Captures narrow artifacts $\Rightarrow$ good on single-source ElevenLabs.
  - Fails on diverse ITW data — overly specific.
- **Ensemble models** (RF, XGBoost — regularized, smooth):
  - Tuned aggressively *against* overfitting (shallow depth, subsampling).
  - Handle ITW diversity better, but miss narrow provider-specific signatures.
- **Linear models** (LR, Linear SVM):
  - Simple decision boundaries — least susceptible to dataset-specific artifacts.
  - Most **consistent** cross-domain generalization.

Key Findings

1. **Feature selection matters:** Mean-only aggregation (195 features) outperformed the full feature set ($\sim$394 features).
2. **Best ITW model:** XGBoost (no mel) — 0.74 accuracy, 0.71 F1, 0.79 AUC on FoR test.
3. **Best generalizers:** Linear models (Logistic Regression, Linear SVM) show the most consistent zero-shot transfer.
4. **No single model dominates** across all zero-shot domains — the choice depends on deployment scenario (narrow vs. diverse).
5. **Mel features are double-edged:** Essential for linear models, but cause overfitting in tree-based models.

Thank you

Questions?

Source code: `https://github.com/g-nazos/audio-deepfake-detection`