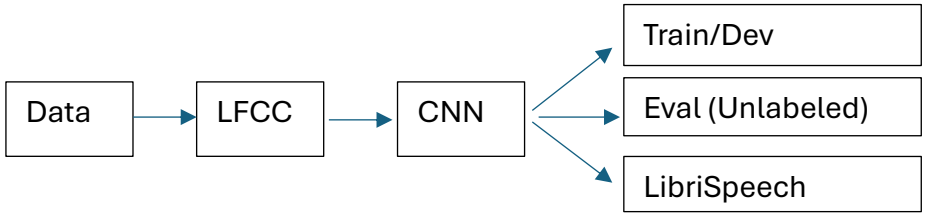
# Evaluating Cross-Dataset Generalization in Speech Spoofing Detection with CNN and LFCC Features


## Abstract

Recent advances in neural text-to-speech and voice conversion have made it possible to generate highly realistic synthetic speech, creating security risks for speaker verification systems, call centers, and any workflow that relies on voice as an authentication factor. This project investigates a convolutional neural network (CNN)–based anti-spoofing model that uses low-frequency cepstral coefficients (LFCC) as its primary feature representation. The model is trained on a spoofing dataset containing both bonafide and synthetic speech in the ASVspoof-style setting, and evaluated across three regimes: (1) in-domain labeled development data, (2) an unlabeled but spoof-rich ASVspoof 2019 LA evaluation split, and (3) purely bonafide out-of-domain human speech drawn from LibriSpeech. On the in-domain development split, the LFCC-CNN achieves near-saturated performance, with AUC ≈ 0.999965, EER ≈ 0.2344%, and accuracy ≈ 99.83%. The model also preserves spoof-detection behavior on the ASVspoof LA evaluation set, predicting 71.96% of trials as spoof, consistent with the spoof-heavy nature of this partition. However, when evaluated on LibriSpeech bonafide speech, the model misclassifies 59.6% of genuine utterances as spoof, revealing a severe domain mismatch and limited cross-corpus generalization. The report describes the data wrangling and exploratory analysis, LFCC feature extraction, CNN architecture evolution from a 3-layer to a 5-layer configuration, training and tuning attempts (including experiments with class weighting, learning rate, and optimizers), and a structured analysis of why the model performs well in-domain but fails on out-of-domain real speech. Limitations are discussed and realistic future directions are outlined.


## Introduction

Neural TTS and voice conversion systems can now generate speech that closely resembles human voices, raising risks for any workflow that relies on voice as an authentication factor. Anti-spoofing modules are therefore increasingly paired with ASV systems to detect synthetic or manipulated audio before speaker verification decisions are made. However, spoofing attacks vary widely, and datasets often represent only a narrow slice of real-world speech diversity. A model may perform extremely well on its training domain yet fail when confronted with new speakers, channels, or spoofing techniques. This project examines that failure mode by training a CNN-based anti-spoofing model on LFCC features and evaluating how well its decision boundary transfers to unseen in-domain data and to genuine speech from an external corpus.

## Problem Definition

The task is framed as binary classification given an audio segment, predict whether it is spoof or bonafide. The central research question is whether a model trained to detect spoofed speech on one dataset will behave sensibly on different speech domains. We examine performance across three evaluation regimes:

(1) labeled in-domain development data

(2) unlabeled but spoof-rich ASVspoof evaluation data

(3) out-of-domain bonafide speech from LibriSpeech

By comparing model behavior in these settings, we assess whether the model learns a general notion of bonafide speech or merely overfits to the spoofing artifacts present in the training dataset.

## Data Source

This project uses three data sources drawn from or aligned with the ASVspoof 2019 LA corpus and LibriSpeech:

The training and development splits for supervised learning come from the ASVspoof 2019 LA dataset, which provides both bonafide and spoofed utterances with ground-truth labels. These splits enable the LFCC-CNN model to learn a discriminative boundary between genuine human speech and synthetic speech generated via TTS or voice conversion systems.

The ASVspoof 2019 LA evaluation split is then used for in-domain testing without labels, as this partition is intentionally released unlabeled for challenge submissions. It is known to contain a high proportion of spoofed trials, making it suitable for assessing whether the model preserves spoof-detection tendencies on unseen but in-domain data.

Finally, a subset of LibriSpeech is used for out-of-domain evaluation. LibriSpeech contains only bonafide human speech from different speakers, channels, and recording conditions than ASVspoof. Because no spoofed speech exists in this corpus, it provides a way to measure false positives—i.e., how often the model incorrectly flags genuine speech as spoof.

Table 1 summarizes the three evaluation roles:

| Dataset / Split | Contains spoof? | Labels available? | Role in project |
|---|---|---|---|
| Train/Dev spoof dataset | Yes (spoof + real) | Yes | Model training and in-domain evaluation |
| ASVspoof LA eval | Yes (spoof-rich) | No (held out) | In-domain, unlabeled evaluation of spoof behavior |
| LibriSpeech subset | No (bonafide only) | All considered real | Out-of-domain evaluation of false positive behavior |

## Data Wrangling

The ASVspoof 2019 LA training and development datasets were prepared for supervised learning through the following steps:

i. **Parsed ASVspoof protocol files**
Extracted trial-level metadata including file identifiers and ground-truth label information defined by the challenge protocols.

ii. **Extracted bonafide vs spoof labels**
Interpreted protocol annotations to distinguish genuine human speech (bonafide) from synthetic/manipulated trials (spoof).

iii. **Converted labels to numeric format**
Encoded labels in binary form for model compatibility:
0 = bonafide (real)
1 = spoof (fake)

iv. **Consolidated train and dev metadata**
Combined protocol outputs from both splits into a unified DataFrame to simplify indexing, partitioning, and batching during modeling.

v. **Attached full audio filepaths**
Resolved .flac filenames to their absolute paths, enabling direct audio loading during feature extraction and training.

vi. **Validated audio file existence**
Verified that all filepaths referenced in the metadata corresponded to actual audio files and flagged missing or mismatched entries.

This consolidated metadata structure served as the foundation for all subsequent preprocessing, feature extraction, and exploratory analysis steps.

Table 2: Metadata Overview

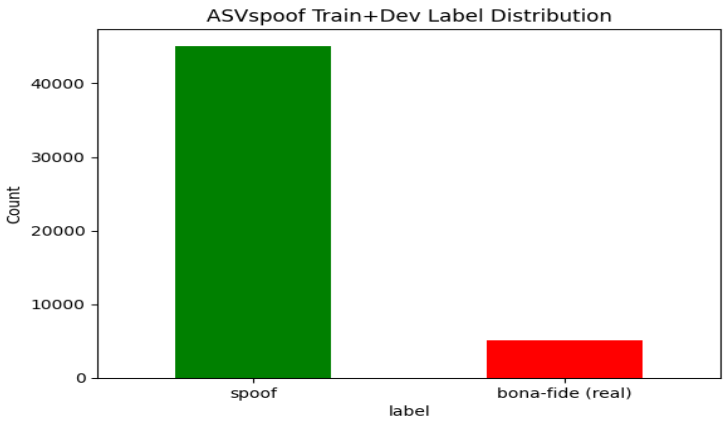
| split | file_id | attack | label | filepath |
|---|---|---|---|---|
| train | LA_T_1138215 | None | 0 | /content/asvspoof/ASVspoof2019_LA_train/flac/L… |
| train | LA_T_1271820 | None | 0 | /content/asvspoof/ASVspoof2019_LA_train/flac/L… |
| train | LA_T_1272637 | None | 0 | /content/asvspoof/ASVspoof2019_LA_train/flac/L… |
| train | LA_T_1276960 | None | 0 | /content/asvspoof/ASVspoof2019_LA_train/flac/L… |
| train | LA_T_1341447 | None | 0 | /content/asvspoof/ASVspoof2019_LA_train/flac/L… |

## Exploratory Data Analysis (EDA)

Exploratory analysis was performed on the combined ASVspoof 2019 LA train and development metadata to understand dataset composition and inform preprocessing decisions. The analysis focused on the following aspects:

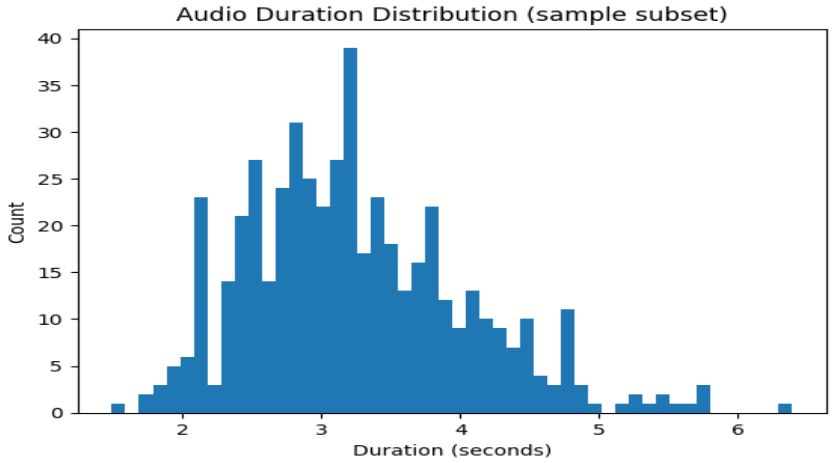
i. **Class balance (bonafide vs spoof)**
The dataset is spoof-heavy, with significantly more spoofed utterances than bonafide speech. This imbalance

has implications for loss weighting and evaluation, as models may receive more gradient signal from spoof examples.
**Result:** Spoof class dominates the distribution.




ii. **Audio duration patterns**

Duration statistics were computed for both classes. Spoofed speech tended to exhibit shorter and more tightly clustered durations, whereas bonafide speech showed longer and more varied durations.
**Summary observation:** bonafide duration distribution right-skewed; spoof distribution concentrated at shorter lengths.



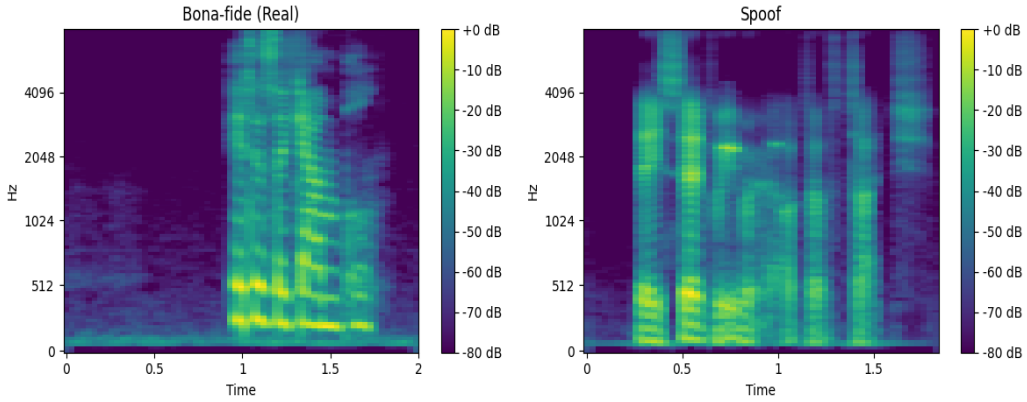

iii. **Basic listening inspection**

Sample audio files were listened to for qualitative understanding. Bonafide speech sounded natural with varied prosody, whereas spoofed audio contained more uniform or machine-generated characteristics depending on the attack type.
**Observation:** human listeners can often distinguish aggressive TTS/VC artifacts, particularly in synthetic voices, even without labels.

iv. **Spectrogram structure**

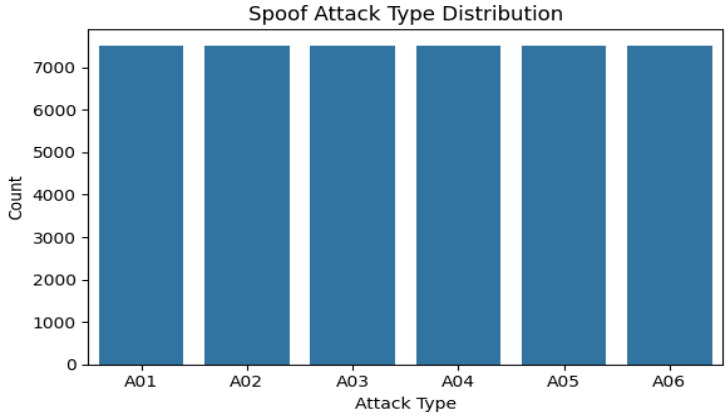
Spectrograms from both classes were visualized. Bonafide speech exhibited richer formant structure with

natural temporal variation, while spoofed audio showed smoother, more uniform spectral profiles. These observations motivated the selection of LFCC features for downstream modeling.




v. **Attack distribution**

Spoof samples in ASVspoof LA originate from multiple TTS/VC systems (referred to as “attacks”). The metadata reveals multiple attack types present across the splits, providing diversity within the spoof class. This supports the use of supervised modeling for detecting synthetic speech artifacts.




Overall, the EDA confirmed that the dataset contains (a) class imbalance, (b) duration asymmetry, and (c) distinct spectral patterns between bonafide and spoof utterances. These findings supported the use of fixed-length input pipelines and LFCC feature extraction in the subsequent modeling stage.

## Preprocessing

The preprocessing pipeline prepared raw ASVspoof LA audio for supervised training by converting it into fixed-format LFCC feature tensors suitable for CNN-based spoof detection. All audio was standardized to a 16 kHz sampling rate and a fixed-duration window to ensure consistent tensor dimensions and prevent duration-based cues from leaking into the classification task. Raw waveforms were resampled and normalized to a common amplitude scale, then either zero-padded or truncated to the target length (4 seconds).

LFCCs (Low-Frequency Cepstral Coefficients) were used as the primary feature representation owing to their sensitivity to subtle spectral artifacts introduced by neural TTS and voice conversion systems. Early experimentation with mel-spectrograms showed weaker separation between bonafide and spoofed speech, motivating the use of LFCCs for the final pipeline. The extraction process used parameters aligned with ASVspoof-style baselines:

– Sampling rate (SR) = 16,000 Hz
– Number of coefficients (N_LFCC) = 60
– FFT size = 512
– Window length = 25 ms (WIN_LENGTH = int(0.025 × SR))
– Hop length = 10 ms (HOP_LENGTH = int(0.010 × SR))

The resulting LFCC maps form a two-dimensional time-by-coefficient representation that integrates naturally with convolutional architectures.

To reduce computational overhead in Colab, LFCC features were precomputed and stored rather than extracted on-the-fly during training. This substantially improved data loading throughput and avoided GPU stalls caused by repeated STFT operations.

The final dataset implementation wrapped these precomputed features in a custom PyTorch Dataset object, which returned (LFCC_tensor, label) pairs for supervised learning. PyTorch DataLoaders were used to batch samples efficiently, shuffle training examples, and provide a consistent interface across the training, development, ASVspoof evaluation, and external LibriSpeech test splits. This modular structure enabled the same feature pipeline to support both in-domain and cross-domain evaluation with minimal changes.

### Baseline LFCC CNN Architecture

The baseline model adopts a lightweight convolutional neural network operating directly on LFCC (Linear Frequency Cepstral Coefficient) feature representations extracted from fixed-length audio waveforms. LFCCs are widely used in ASV and anti-spoofing research due to their ability to preserve fine spectral detail, particularly in regions where vocoder and neural synthesis artifacts often manifest. In contrast to mel-based representations, which emphasize perceptual smoothness, LFCCs retain sharper spectral transitions that support synthetic artifact discrimination.

Each input sample is represented as a tensor of shape **[1 × 60 × T]**, corresponding to a single-channel cepstral map with 60 cepstral coefficients across time. The network processes this representation through a sequence of convolutional blocks consisting of **Conv → BatchNorm → ReLU**, allowing local spectral structures to be captured and combined hierarchically. Early layers employ **max pooling** to reduce temporal resolution and aggregate information, while later layers increase channel depth (e.g., **16 → 32 → 64 → 128**) to support richer feature abstraction as the receptive field expands.

Following the convolutional stack, the feature map is flattened and passed through a small fully connected head that outputs a single logit for **binary classification (spoof vs bonafide)**. The final layer is trained using **BCEWithLogitsLoss**, which provides a numerically stable formulation for the combination of sigmoid activation and binary cross-entropy.

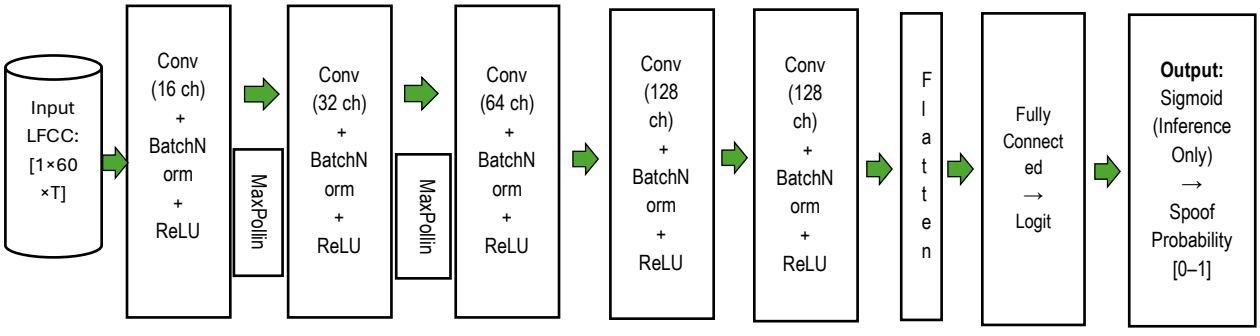
This LFCC-CNN configuration is intentionally compact to avoid over-parameterization and support efficient inference, while retaining sufficient expressiveness to capture spoof-related spectral artifacts.

### Early 3-Layer CNN Attempt and Failure Modes

Prior to settling on the final architecture, a shallower **3-layer CNN** was implemented using the same LFCC input format. Although the shallow model demonstrated that LFCCs were informative and enabled basic separation between spoof and bonafide speech, it exhibited several limitations:

- **Insufficient capacity:** the shallow network lacked depth to model higher-order spectral dependencies characteristic of TTS/VC artifacts.
- **Weak hierarchical abstraction:** without additional convolutional blocks, the network relied heavily on local filters and failed to integrate longer-range temporal structure.
- **Training instability on development split:** the model converged inconsistently and did not match the validation performance achieved later by the deeper configuration.

These failure modes motivated the shift to a deeper **5-layer convolutional backbone**, which improved representational capacity without significantly increasing inference cost. The 5-layer model ultimately produced the in-domain results presented later in the evaluation section.




Baseline LFCC CNN Architecture

## Training and Validation

Model training was conducted on the ASVspoof LA training split using binary supervision, while the development split was reserved for validation. The dataset exhibits a strong class imbalance, with spoofed utterances appearing far more frequently than bonafide ones. To address this imbalance, the loss function was weighted using the pos_weightparameter in BCEWithLogitsLoss. In PyTorch, pos_weight scales the loss contribution of the **positive class** and is defined as the ratio:

*pos_weight = N_negative / N_positive*

In this project, bonafide speech was treated as the positive class and was substantially less frequent than spoofed speech. The correct ratio was therefore approximately **8.8**, reflecting that there were roughly 8.8 spoof samples for every bonafide sample. Using the inverse ratio (**0.11**) would incorrectly down-weight the minority class and worsen the imbalance by encouraging the model to ignore bonafide errors. Empirically, the 8.8 weighting improved sensitivity to bonafide speech during validation without destabilizing training.

Optimization was performed using the **Adam** optimizer, chosen for its fast convergence and robustness under LFCC feature input. Alternative optimizers such as SGD were not pursued extensively due to their sensitivity to learning rate schedules. Several learning rates were tested; higher learning rates caused oscillatory validation behavior, while very low values slowed convergence with no compensatory gains. The selected learning rate provided reliable convergence under the 5-layer CNN configuration.

A learning rate scheduler (**ReduceLROnPlateau**) was also incorporated. This scheduler monitors a target metric—in this case, validation loss—and automatically reduces the learning rate when improvement stalls. The rationale is that large learning rates are beneficial during early optimization, but finer adjustments are required as the model approaches a minimum. ReduceLROnPlateau thus enables more stable late-stage refinement without manual retuning and can prevent premature stagnation or over-stepping in narrow minima.

During training, model parameters were updated via mini-batch gradient descent, and development performance was monitored each epoch to detect overfitting or collapse. Under the final tuning configuration, the model displayed smooth loss behavior and strong alignment between training and development curves, indicating that the optimization regime successfully learned an in-domain decision boundary for spoof detection.

Summary of key tuning experiments

| Aspect | Variants tested | Observation | Final choice |
|---|---|---|---|
| pos_weight | No weighting, moderate, aggressive | Aggressive weighting destabilized training | None or moderate, depending on run |
| Learning rate | Lower, baseline, higher | Too high → divergence; too low → slow convergence | Baseline rate with stable convergence |
| Optimizer | Adam, possibly SGD/others | Adam converged faster and more stably | Adam |

## In-Domain Evaluation on Labeled Development Data

The LFCC-CNN model was trained for 10 epochs on the ASVspoof LA training split and evaluated on the labeled development split. Training converged rapidly, with the training loss decreasing from **0.537** → **0.0005** within 10 epochs. The development loss reached **0.0076**, indicating strong in-distribution generalization under matched training and evaluation conditions. A brief increase in development loss was observed around epochs 4–5, a temporary fluctuation commonly encountered in imbalanced binary classification tasks, particularly when batch composition is spoof-heavy. After this point, the development loss resumed its downward trend and remained low for the remainder of training. There were no signs of underfitting, as the model quickly learned meaningful LFCC patterns associated with spoof artifacts, while only mild overfitting appeared toward the end of training as the training loss continued to decrease after the development loss plateaued.

The following metrics were computed:

- AUC (Area Under the ROC Curve)
- EER (Equal Error Rate)
- Accuracy (at a threshold of 0.5)

The implementation for AUC and EER used scikit-learn's roc_curve and roc_auc_score functions. EER was computed by finding the point on the ROC curve where false positive rate (FPR) and false negative rate (FNR) are approximately equal.

Quantitatively, the model achieved near-saturated anti-spoofing performance on the development split:

- **AUC:** 0.999965
- **EER:** 0.2344%
- **Accuracy:** 99.83%

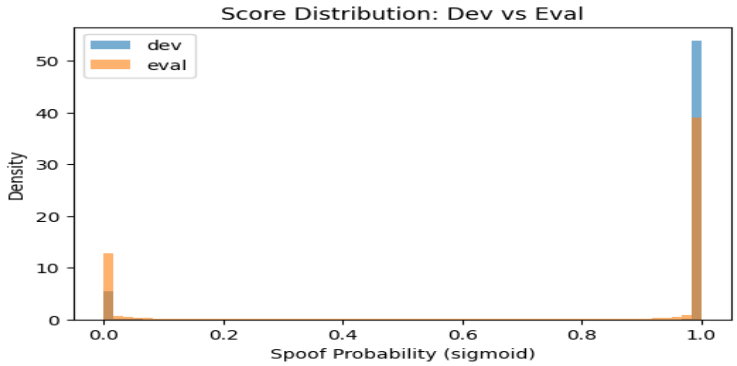
These metrics confirm that the model learned a highly discriminative decision boundary for the ASVspoof LA task. Additionally, approximately **89.91%** of development utterances were predicted as spoof at a 0.5 threshold, a value consistent with the spoof-heavy class distribution of the development split. Taken together, the results demonstrate that the network exhibits excellent in-domain classification capability and strong exploitation of LFCC spectral cues associated with spoofing.

### In-Domain Evaluation on Unlabeled ASVspoof LA Evaluation Split

To assess whether the model's spoof-detection behavior transferred to unseen in-domain audio, the trained LFCC-CNN was applied to the ASVspoof 2019 LA evaluation split. Unlike the train and development partitions, ground-truth labels for the evaluation split are not released, preventing the computation of AUC, EER, or accuracy. However, the evaluation set is known to be spoof-rich, and model behavior can be examined through the distribution of spoof probabilities and decision outputs.

On the development split, **89.91%** of utterances were classified as spoof at a 0.5 threshold, reflecting the spoof-heavy class distribution. On the evaluation split, the model predicted **71.96%** of utterances as spoof, again indicating a spoof-leaning decision boundary under unseen evaluation conditions. Importantly, the model did not collapse on the evaluation split: it continued to produce confident scores that separated trials into low-spoof and high-spoof regions rather than converging toward a constant output or 0.5. In other words, the decision function remained operational, and the model successfully produced usable 0/1 classifications even though the true accuracy remains unknown without labels.

While accuracy may vary depending on the true label distribution, the preservation of meaningful binary decision behavior demonstrates that the LFCC-CNN generalizes within the ASVspoof domain and does not degrade into a degenerate classifier when exposed to evaluation-grade conditions.




Dev VS Eval Score Distribution

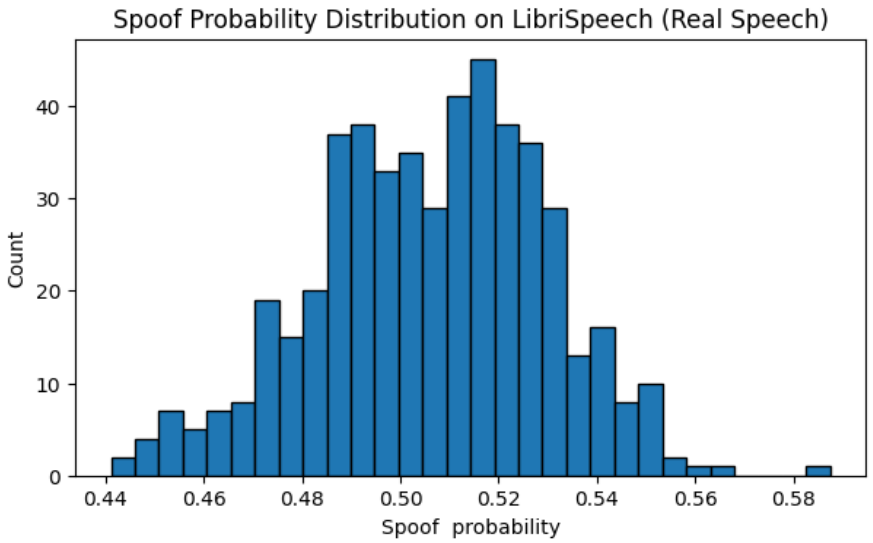
## Out-of-Domain Evaluation on LibriSpeech Bonafide Speech

To examine cross-corpus generalization, the trained LFCC-CNN was evaluated on a subset of the LibriSpeech corpus containing only bonafide human speech. LibriSpeech differs substantially from ASVspoof in speaker identity, linguistic content, channel characteristics, and recording conditions, making it an appropriate test of whether the model's decision boundary extends beyond the distribution it was trained and validated on.

The exact same preprocessing pipeline used for ASVspoof was applied to LibriSpeech, including waveform normalization, fixed-duration padding/truncation, and LFCC feature extraction. LFCCs were again precomputed to ensure that any performance degradation could not be attributed to feature quality or data loader inefficiency. This alignment ensured that evaluation differences stemmed from dataset shift rather than preprocessing mismatch.

Because LibriSpeech contains no spoofed utterances, standard anti-spoofing metrics such as EER or AUC cannot be computed directly. Instead, the relevant measure is the false positive rate—how often genuine speech is misclassified as spoof. At a 0.5 threshold, the LFCC-CNN predicted **59.6%** of LibriSpeech utterances as spoof. Although ground-truth spoof labels are absent, the model did not collapse or saturate; it continued to emit non-degenerate probability scores and maintained a functional decision boundary rather than defaulting to a constant output.

We further examined the score distribution through histogram analysis of spoof probability. The distribution revealed a heavy skew toward high spoof scores, confirming that unfamiliar bonafide speech was frequently flagged as synthetic.




A brief speaker-level aggregation demonstrated that this behavior was consistent across speakers rather than confined to a single talker, indicating a broad generalization gap rather than idiosyncratic speaker mismatch.

| **Speaker ID** | **1221** | **7729** | **2300** | **260** | **1089** | **5639** | **7021** | **7176** | **8224** | **121** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Spoof Rate** | 1.0000 | 0.8889 | 0.8333 | 0.8235 | 0.8182 | 0.7778 | 0.7778 | 0.7500 | 0.7500 | 0.7333 |

Some speakers trigger much higher spoof scores than others. for example, several speakers show false positive rates above 0.80 (e.g., speaker 1221: 1.00, 7729: 0.89, 2300: 0.83), even though all clips are real speech. this indicates that false positives are not random but depend on who is speaking.

Taken together, these results suggest that while the LFCC-CNN generalized well within the ASVspoof domain, its notion of "genuine human speech" remained dataset-specific. The model learned to detect particular spectral artifacts characteristic of ASVspoof spoofing systems but tended to over-flag natural speech outside that domain as synthetic. This highlights the practical distinction between detecting known spoofing attacks and authentically validating human speech under distribution shift.

### Comparison with Original ASVspoof Performance

On the ASVspoof2019-LA benchmark, the model performed extremely well on the in-distribution train and development splits:

- AUC: 0.999965
- EER: 0.2344%
- Accuracy: 99.8309%

These results indicate near-perfect separation between bona fide and spoofed samples within the ASVspoof domain.

When evaluating the benchmark's unlabeled development and evaluation splits using predicted spoof fractions, the model estimated:

- Development predicted spoof: 89.91%
- Evaluation predicted spoof: 71.96%

These values are expected because both splits contain a mixture of bona fide and synthetic voice samples, and the spoofing attacks follow controlled and consistent artifact patterns.

In contrast, when the same model was tested on the LibriSpeech test-clean dataset, which contains only real human speech from multiple speakers and recording conditions, the model incorrectly classified 59.6% of the samples as spoof. This reflects a substantial increase in false positives under normal speech variability and domain shift.

| Evaluation Setting | Labels Available | Metric Type | Result |
|---|---|---|---|
| **ASVspoof LA (Dev)** | Yes | AUC / EER / Accuracy | **AUC:** 0.999965 • **EER:** 0.2344% • **ACC:**99.8309% |
| **ASVspoof LA (Eval)** | No | Predicted Spoof Fraction | **71.96%** spoof predicted |
| **LibriSpeech (Test-Clean)** | No (bonafide only) | False Positive Rate | **59.6%** FPR (bonafide misclassified as spoof) |

Cross Domain Comparison

## Why the Model Failed to Generalize

Although both ASVspoof-LA and LibriSpeech contain genuine human speech, they originate from fundamentally different domains. The LFCC-CNN model learned to discriminate spoofed speech from bona fide speech within the ASVspoof distribution but did not learn a general notion of "real human speech." Instead, it internalized a narrower decision rule tied to the specific acoustic and structural patterns present in ASVspoof's bona fide subset. When tested on LibriSpeech, this decision rule failed to transfer, leading to high false positive rates under cross-corpus evaluation.

Several factors contributed to this generalization failure:

**Domain mismatch.** ASVspoof bona fide audio is controlled in style, speaking rate, and channel characteristics, while LibriSpeech originates from audiobook narration. LibriSpeech speech exhibits expressive prosody, varied pacing, and natural articulation patterns that differ from the conversational/controlled setting in ASVspoof. The model had never encountered these patterns during training.

**Speaker variability.** ASVspoof contains a relatively limited set of speakers compared to LibriSpeech, which includes hundreds of narrators with diverse accents, timbre, and vocal physiology. Because speaker diversity directly influences spectral and cepstral patterns, the model's exposure to only one bona fide speech domain limited its ability to map the full range of genuine speech variability.

**Recording and channel differences.** ASVspoof recordings tend to be made under more uniform acoustic and hardware conditions, whereas LibriSpeech includes different microphones, production pipelines, levels of post-processing, and studio environments. Anti-spoofing models are highly sensitive to channel characteristics and shifts in channel distributions can be misinterpreted as spoof artifacts.

**Prosody and phonetic structure.** Audiobook narration involves expressive intonation, elongated vowels, and clear articulation designed for intelligibility. These prosodic differences influence LFCC structure and can superficially resemble spectral smoothing effects seen in some spoofing systems, especially to an LFCC-based classifier.

**Artifact overfitting.** The LFCC-CNN benefited from strong within-benchmark spoof artifacts, leading to near-perfect ASVspoof performance but implicitly encouraging overfitting to attack-specific cues rather than learning a general "bona fide human speech prior." Once those cues disappeared, benign variability was misinterpreted as synthetic.

**Limited bona fide diversity.** Crucially, the model only ever saw one bona fide speech distribution during training—ASVspoof's. Without exposure to other bona fide speech corpora, the model developed an overly narrow boundary for "real," making unfamiliar real speech appear anomalous and therefore "spoof-like."

**Calibration and threshold shift.** Finally, score calibration learned under ASVspoof does not transfer to LibriSpeech. The model's sigmoid outputs remain separable and non-degenerate, but the score distribution is shifted, causing the ASVspoof-calibrated 0.5 threshold to produce excessive false positives.

Collectively, these factors caused unfamiliar bona fide speech to be systematically misclassified as spoof. The model succeeded at detecting spoof patterns it had seen before, but failed to recognize genuine human speech beyond the controlled domain it was trained to understand.

## Practical Implications

These results show that high anti-spoofing performance on a controlled benchmark does not guarantee that a system will correctly accept legitimate users in real deployment scenarios. A model that rejects unfamiliar bona fide speech—as observed on LibriSpeech—would severely undermine usability in voice authentication, speaker verification, or fraud prevention pipelines. In practice, such a system would increase false rejections of genuine users, creating operational failures even if spoof detection accuracy appears excellent on paper. This highlights a critical gap between benchmark performance and real-world acceptance criteria: anti-spoofing models must generalize to diverse speaker populations, recording channels, and speaking styles in addition to detecting spoofing artifacts.

## Future Work

There are a few realistic next steps that build directly on this project:

- **Evaluate on ASVspoof Eval with Ground Truth:**
  A natural next step is to obtain the ASVspoof evaluation labels to measure how well the model handles unseen attacks within the ASVspoof domain. This was not done here because the eval split is unlabeled by default.
- **Test on More Real Speech Datasets:**
  Evaluating the model on other real speech datasets (e.g., Common Voice, VoxCeleb) would help determine whether the failure on LibriSpeech is dataset-specific or a broader generalization issue.
- **Increase Bona Fide Diversity:**
  Training with more diverse real speech could help the model learn a more general notion of bona fide speech and reduce false positives.
- **Adjust Threshold Calibration:**
  Tuning the decision threshold or calibrating scores could reduce unnecessary false positives in practical settings.

These steps focus on improving evaluation and understanding without changing the model architecture.

## Conclusion

This project trained a CNN-based anti-spoofing model using LFCC features and evaluated its performance across three regimes: in-domain labeled development data, in-domain unlabeled ASVspoof 2019 LA evaluation data, and out-of-domain bonafide speech from LibriSpeech. On the in-domain development split, the model performed extremely well, with AUC ≈ 0.999965, EER ≈ 0.2344%, and accuracy ≈ 99.83%, and correctly reflected the spoof-heavy class distribution with 89.91% of dev trials predicted as spoof. When applied to the unlabeled ASVspoof evaluation split, the model continued to behave as a spoof detector, predicting 71.96% of trials as spoof, consistent with the spoof-rich nature of that partition.

However, when evaluated on bonafide speech from LibriSpeech, the model misclassified 59.6% of genuine utterances as spoof, revealing a severe domain mismatch and limited cross-corpus generalization. This shows that an anti-spoofing model that appears nearly perfect on its home dataset can still fail dramatically when exposed to different recording conditions and speaker populations.

From a learning perspective, the project demonstrates the full pipeline of a modern audio ML experiment: data wrangling, EDA, fixed-length preprocessing, LFCC feature extraction, CNN architecture design (from 3-layer to 5-layer), hyperparameter tuning, and multi-regime evaluation. From a research perspective, it highlights an important reality of anti-spoofing systems: strong in-domain performance does not guarantee that models will behave correctly on real-world speech. Addressing this gap will require broader training data, better feature design, and more systematic domain adaptation, all of which provide clear directions for future work.

## References


***ASVspoof Benchmark & Anti-Spoofing***

*Todisco, M., Wang, X., Vestman, V., Sahidullah, M., Kinnunen, T., & Evans, N. (2019).*
*ASVspoof 2019: A large-scale public database of synthetic, converted and replayed speech.*
*In:* ***Interspeech 2019****.*

***LFCC Feature Justification***

*Sahidullah, M., Kinnunen, T., & Hanilçi, C. (2015).*
*Robust detection of synthetic speech using Linear Frequency Cepstral Coefficients.*
***IEEE/ACM Transactions on Audio, Speech, and Language Processing****.*

*Reima, F., et al. (2020).*
*Linear Frequency Cepstral Coefficients for Replay Spoofing Detection.*
*In:* ***Odyssey 2020*** *Speaker and Language Recognition Workshop.*

***LibriSpeech (Bonafide Out-of-Domain Corpus)***

*Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015).*
*Librispeech: An ASR corpus based on public domain audiobooks.*
*In:* ***ICASSP 2015****.*

***Cross-Corpus Generalization / Domain Shift***

*Soni, R., Patil, A., & Patil, H. (2021).*
*A study on cross-corpus generalization for spoofing detection.*
***Computer Speech & Language****.*

*Bhattacharjee, A., Sahidullah, M., & Saha, G. (2020).*
*Spoofing and anti-spoofing under domain shift: A comprehensive evaluation.*
*In:* ***Interspeech 2020****.*

**Tooling (Training Framework)** — optional but legitimate in academic reports

Paszke, A., et al. (2019).
*PyTorch: An Imperative Style, High-Performance Deep Learning Library.*
In: **NeurIPS 2019**.