THE UNIVERSITY OF SYDNEY

Project Report: Detection of Voice Deepfakes Using LFCC and MFCC Features with GMM and CNN Models

ELEC5305: Acoustics, Speech and Signal Processing

GitHub Project Link with Video:
https://github.com/nmalameri/elec5305-project-510044280.git

Nasser Alameri, 510044280

FACULTY OF ENGINEERING

Contents

1 Abstract

Voice deepfakes generated by modern text-to-speech and voice conversion systems pose a growing threat to automatic speaker verification (ASV) and downstream applications. This project implements a lightweight, fully reproducible deepfake speech detector following the ASVspoof 2019 Logical Access (LA) protocol. The system classifies utterances as bona fide or spoofed using compact cepstral features; linear frequency cepstral coefficients (LFCC) and mel frequency cepstral coefficients (MFCC); paired with two simple classifiers: Gaussian mixture models (GMMs) and a shallow convolutional neural network (CNN). All models are designed for CPU-only inference and are evaluated with equal error rate (EER) under a DEV$\rightarrow$EVAL fixed-threshold setup. On the ASVspoof 2019 LA evaluation set, LFCC + GMM achieves the best EER of 9.57%, while MFCC + GMM attains the highest accuracy of 84.12%. The CNN yields near-perfect separation on the development split but generalises less robustly to evaluation, highlighting sensitivity to distribution shift and crop choices. Pretrained waveform and graph-based models (RawNetLite, RawNet2, AASIST-LA) perform markedly worse under this protocol, underscoring the importance of matched preprocessing and domain conditions. Overall, the results show that carefully implemented cepstral front ends with simple generative classifiers remain competitive, interpretable, and resource-efficient baselines for deepfake speech detection.

2 Introduction

Voice deepfakes are synthetic speech signals generated by modern text to speech (TTS) and voice conversion (VC) systems that can sound increasingly realistic. Their misuse poses risks ranging from identity fraud and misinformation to subverting automatic speaker verification (ASV); this has motivated community benchmarks such as the ASVspoof challenges aimed at designing and evaluating countermeasures that protect ASV from manipulation [1].

This project builds a lightweight, reproducible deepfake speech detector in Python. Given an input utterance, the system decides bona fide vs. spoofed using compact spectral features; Linear Frequency Cepstral Coefficients (LFCC) and Mel Frequency Cepstral Coefficients (MFCC); paired with Gaussian Mixture Model (GMM) and shallow Convolutional Neural Network (CNN) classifiers. These design choices align with established ASVspoof baselines that prioritise simplicity and interpretability [2]. Our contribution is a fully reproducible, CPU friendly baseline that systematically contrasts LFCC vs. MFCC with GMM and a shallow CNN; this clarifies how far compact models can go on LA without heavy architectures.

I use the ASVspoof 2019 Logical Access (LA) corpus, which contains bonafide speech and spoofed utterances generated with contemporary TTS/VC systems; this enables controlled and reproducible evaluation of spoofing countermeasures [2]. For evaluation, I report the Equal Error Rate (EER) for standalone countermeasure performance [3].

Finally, I assess whether LFCC or MFCC offers stronger discriminative power. Prior work reports that LFCC can outperform MFCC for synthetic speech detection on certain benchmarks; a plausible reason is its linear frequency emphasis that preserves higher frequency details relevant to artefacts; outcomes remain dataset dependent [4].

Research Question. On ASVspoof 2019 LA, can LFCC+GMM and a shallow CNN; implemented for CPU-only inference; achieve $\leq$ 10–15% EER on the EVAL set when using a DEV-set fixed decision threshold, and how do they compare to MFCC baselines in accuracy and efficiency?

Recent anti-spoofing work has also explored waveform and graph based models such as RawNetLite, RawNet2, and AASIST; which are more complex than traditional cepstral baselines and usually rely on specialised preprocessing pipelines. Although reproducing their full training setup is outside the scope of this project, evaluating their pretrained versions helps place the lightweight LFCC/MFCC models in a broader research context and shows how simple baselines compare with more advanced architectures and have therefore been implemented as such.

3 Literature Review

This section summarises prior work that motivates the design choices in this project and clarifies how the community evaluates progress on deepfake speech detection. The review is organised around four themes: benchmark tasks and datasets; feature representations; classifier families; and evaluation metrics and protocols.

3.1 Benchmark context and datasets

The ASVspoof initiative provides a common framework for the study of synthetic and deepfake speech detection with standardised corpora, protocols, and baselines. The first challenge in 2015 introduced a shared evaluation for text to speech and voice conversion attacks and set the template for subsequent editions [5]. The 2017 challenge focused on replay attacks and documented the difficulty of generalising across devices and acoustic conditions [6, 7]. The 2019 edition released large scale logical access and physical access databases together with reference countermeasures and scoring protocols, consolidating the field benchmark used in this project [2]. The 2021 edition added a dedicated deepfake task and reinforced good practice on metrics and reporting while introducing channel and compression variability to stress generalisation [1]. Public challenge pages document baseline resources and post challenge releases for reproduction [8, 9].

3.2 Feature representations

Cepstral features derived from short term spectra remain a strong choice for synthetic speech detection because they capture spectral envelope structure with compact dimensionality. Mel frequency cepstral coefficients (MFCC) use a mel spaced filterbank that emphasises perceptual low frequency resolution, whereas linear frequency cepstral coefficients (LFCC) retain more detail at higher frequencies [10]. Comparative evidence shows that linear spacing can expose artifacts introduced by synthesis or conversion that may be smoothed on the mel scale; this often leads to better discrimination between bona fide and spoofed speech, though results are dataset dependent [4]. Constant Q cepstral coefficients extend this idea by using the constant Q transform before cepstral processing and became a widely used baseline in early challenges [11].

3.3 Classifier families

Two complementary approaches have been influential. Generative models such as Gaussian mixture models fit separate densities to bona fide and spoof feature vectors and use the log likelihood ratio as a detection score; this approach is simple and efficient and is part of the official ASVspoof baselines when paired with LFCC or CQCC features [2, 12]. Discriminative neural models learn local time frequency patterns indicative of spoofing from cepstral sequences or spectrogram like inputs. Representative systems include lightweight convolutional networks used by top systems in ASVspoof 2019 [13] and the AASIST architecture which integrates spectro temporal graph attention; a compact variant (AASIST–L) uses only $\sim$85k parameters [14]. Other competitive lightweight designs combine convolutional front ends with recurrent or residual modules, such as LC–GRNN [15, 16].

3.4 Evaluation metrics and protocols

Equal error rate is widely reported for countermeasure evaluation in ASVspoof papers and follow up studies; however, to reflect deployment with speaker verification systems, ASVspoof 2019 introduced the tandem detection cost function which jointly assesses a countermeasure and an automatic speaker verification system under a common cost model [3, 17]. While this report emphasises equal error rate for comparability with many baselines, I follow the established protocol conventions of strict train, development, and evaluation separation to avoid optimistic bias [2].

3.5 Summary and implications for this project

The reviewed literature supports three design choices. First, the ASVspoof 2019 logical access corpus offers a well defined and widely used testbed with clear protocols and established baselines [2]. Second, linear frequency cepstral coefficients are expected to reveal high frequency synthesis artefacts more clearly than mel frequency cepstral coefficients in many cases [4]. Third, compact classifiers from both the generative and discriminative families are appropriate and can be implemented efficiently while maintaining strong detection accuracy [13, 14].

More recent models, including RawNet2 and AASIST, introduce stronger deep-learning architectures but depend heavily on their original waveform preprocessing. Including these pretrained systems in this project helps position the cepstral baselines alongside modern approaches.

4 Methodology

This section details the end to end pipeline; corpus protocol; feature extraction; classifiers (GMM and a lightweight CNN); training procedures; and evaluation. Choices were made to be reproducible, CPU and MPS friendly, and directly comparable across LFCC and MFCC.

4.1 Data and protocol

Corpus ASVspoof 2019 Logical Access with the official countermeasure protocols [2, 3].

Splits I use the train; development; and evaluation partitions as released; per the countermeasure protocols there is no speaker or file leakage across splits [3].

Sampling rate All audio is resampled to $f_s = 16\,\text{kHz}$ to match Logical Access specifications [2].

Task Binary spoofing detection with two classes: *bona fide* and *spoof*.

Evaluation policy Thresholds are tuned on development at the minimum equal error rate operating point and then fixed when scoring evaluation; no retuning on evaluation.

4.2 Pre processing and features

Framing 25 ms Hann windows with 10 ms hop. For frame index t,

$$X_t[k] \;=\; \sum_{n=0}^{L-1} x[n+tH]\, w[n]\, e^{-j2\pi kn/L}, \qquad P_t[k] \;=\; |X_t[k]|^2 .$$

Filterbanks Let $F_m[k]$ be a non negative triangular filter.

- LFCC: M linearly spaced filters over $[0, f_s/2]$;
- MFCC: M mel spaced filters.

Filterbank energies:

$$E_t(m) \;=\; \sum_k P_t[k]\, F_m[k], \qquad m = 1, \ldots, M .$$

Cepstra After log compression with $\varepsilon > 0$,

$$c_t(n) \;=\; \sum_{m=1}^{M} \log\big(E_t(m) + \varepsilon\big)\, \cos\Big(\tfrac{\pi n}{M}\big(m - \tfrac{1}{2}\big)\Big), \qquad n = 0, \ldots, N_c - 1 .$$

I keep c_0 and use $N_c \approx 20$ for both LFCC and MFCC with Δ and Δ^2; compare with LFCC and CQCC baselines in ASVspoof [12].

Dynamics First and second order deltas via symmetric regression with half window S:

$$\Delta c_t \;=\; \frac{\sum_{s=1}^{S} s\,(c_{t+s} - c_{t-s})}{2\sum_{s=1}^{S} s^2}, \qquad \Delta^2 c_t \text{ from } \Delta c_t \text{ analogously.}$$

Standardisation Global z score per dimension using train statistics: $\tilde{x} = (x - \mu)/\sigma$; the same scaler is reused on development and evaluation to prevent leakage. For the CNN, inputs are these globally standardised features; no per utterance re normalisation.

4.3 Models

4.3.1 GMM baseline

Model Two diagonal covariance Gaussian mixture models trained at the frame level; one per class $y \in \{\text{bona}, \text{spoof}\}$:

$$p(x \mid M_y) \;=\; \sum_{k=1}^{K} \pi_k^{(y)}\, \mathcal{N}\big(x \mid \mu_k^{(y)}, \Sigma_k^{(y)}\big), \quad \Sigma_k^{(y)} = \operatorname{diag}(\sigma^2)\,.$$

This follows established countermeasure baselines used in ASVspoof [2, 12].

Training Expectation–Maximisation on pooled frames with compute safe caps. With posteriors

$$\gamma_{tk}^{(y)} \;=\; \frac{\pi_k^{(y)}\, \mathcal{N}(x_t \mid \mu_k^{(y)}, \Sigma_k^{(y)})}{\sum_j \pi_j^{(y)}\, \mathcal{N}(x_t \mid \mu_j^{(y)}, \Sigma_j^{(y)})},$$

updates are

$$N_k = \sum_t \gamma_{tk}, \quad \mu_k = \frac{1}{N_k}\sum_t \gamma_{tk} x_t, \quad \Sigma_k = \frac{1}{N_k}\sum_t \gamma_{tk}(x_t - \mu_k)^2, \quad \pi_k = \frac{N_k}{\sum_j N_j}\,.$$

Practical controls: $K \in \{64, 128\}$; diagonal covariance; k means initialisation; maximum iterations with tolerance; per utterance frame cap for example 100 to 200 frames and a spoof utterance cap for example 6000 with balanced subsampling across attack IDs A01 to A06 to avoid over representing any single spoof type. These capping and sampling choices are engineering pragmatics for fair; compute bounded comparison.

Scoring Utterance scores are mean log likelihood ratio over its T_u frames:

$$s(u) \;=\; \frac{1}{T_u}\sum_{t=1}^{T_u}\Big[\log p(x_t \mid M_{\text{spoof}}) - \log p(x_t \mid M_{\text{bona}})\Big]\,.$$

4.3.2 CNN baseline

Inputs Fixed length time and cepstral maps with $T = 400$ frames; cropped centrally on development and evaluation and randomly on train. The tensor shape is $[1, T, D]$ with $D \approx 60$ when deltas are included. Zero padding is applied if an utterance is shorter than T.

Architecture A compact two dimensional CNN; three convolution; batch normalisation; ReLU blocks with 3×3 or 5×5 kernels and two 2×2 max pools; followed by global average pooling over T and D and a linear head that outputs a single logit z. This keeps parameters and FLOPs low while capturing local time and frequency patterns; see lightweight LCNN and AASIST lines of work [13, 14].

Loss and optimisation Weighted binary cross entropy on logits,

$$\mathscr{L} \;=\; -w_1 y \log \sigma(z) \;-\; w_0 (1-y) \log\big(1-\sigma(z)\big),$$

with $w_1/w_0 \approx N_{\text{bona}}/N_{\text{spoof}}$ from the possibly capped train set. Optimiser: AdamW with cosine learning rate schedule. Device: CPU; Apple MPS; or CUDA when available. To ensure a fair comparison with GMM, I apply the same spoof utterance cap and attack balanced sampling.

4.3.3 Pretrained Waveform Models

To extend the baseline system and place the cepstral models in a broader context, three pretrained anti-spoofing architectures were also evaluated: RawNetLite, RawNet2, and AASIST-LA. These models were originally trained on the ASVspoof 2021 LA/DF challenge dataset, giving them exposure to a wide range of modern synthesis and voice conversion attacks. Including these systems provides an additional comparison point and allows this project to examine how larger pretrained models behave under the ASVspoof 2019 LA protocol.

For practical and computational reasons, the full training pipelines used for these models (specific waveform pre-emphasis, sinc-based filtering, and model-dependent normalisation) were not reproduced. Instead, a simplified preprocessing wrapper was used so that all pretrained models could be run within the same codebase and evaluated under the same DEV→EVAL thresholding setup; making the comparison consistent and highlighting the effect of preprocessing differences on pretrained models, while still showing how the lightweight cepstral systems perform relative to more advanced architectures trained on larger and more recent datasets.

4.4 Operating point and metrics

From the pairs $\{(s_i, y_i)\}$ I compute ROC curves and the equal error rate at threshold $\tau^\star$ where

$$\text{FPR}(\tau^\star) \;=\; \text{FNR}(\tau^\star) \;=\; 1-\text{TPR}(\tau^\star).$$

I report development and evaluation equal error rate; at the development chosen threshold I also report accuracy and balanced accuracy,

$$\text{ACC} = \frac{TP+TN}{N}, \qquad \text{BACC} = \tfrac{1}{2}\big(\tfrac{TP}{TP+FN} + \tfrac{TN}{TN+FP}\big),$$

as well as TPR and FPR. For analysis I compute per attack equal error rate on evaluation by grouping scores using the provided attack IDs A07 to A19 [2,3].

4.5 Experimental controls

Feature settings

LFCC: $M = 70$; $N_c = 20$; c_0 kept; Δ and Δ^2 on. MFCC: $M = 80$; $N_c = 20$; c_0 kept; Δ and Δ^2 on.

Values were tuned on development; choices are recorded in each run configuration.

Pooling for GMM Cap about 100 to 200 frames per utterance; spoof cap for example 6000 balanced over attack IDs to mitigate sampling bias.

CNN training Crop length $T = 400$; batch size and epochs per run configuration; AdamW learning rate and positive class weight printed in logs.

Randomness Fixed seeds for subsampling; scaler fitting; shuffling; and model initialisation.

4.6 Reproducibility and artefacts

Each run creates a dedicated folder under `results/models/{gmm_lfcc, gmm_mfcc, cnn_lfcc, cnn_mfcc}/<tag>` that stores:

- `run_config.json` with all hyperparameters and environment;
- the fitted scaler; GMM parameters in `.joblib` or CNN weights in `model.pt`;
- per utterance scores `scores_{dev,eval}.csv`;
- ROC samples `roc_{dev,eval}.csv` and figures;
- per attack tables with equal error rate and operating point metrics;
- a summary `metrics.json` with development equal error rate and threshold; and evaluation equal error rate; accuracy; TPR; FPR; and balanced accuracy;

Run tags encode key choices; for example `K64_cap100_S6000_F800000_YYYYMMDD_HHMM` for GMM or `T400_E15_B32_S6000_BAL_YYYYMMDD_HHMM` for CNN; enabling side by side comparison and exact reproduction.

5 Results

We evaluate on the ASVspoof 2019 LA protocol; the operating threshold is chosen on the development (DEV) split at the minimum EER and then applied unchanged to the evaluation (EVAL) split. We compare LFCC and MFCC paired with a GMM and a compact CNN; we also analyse attack wise behaviour. In addition, we include pretrained waveform-based models (RawNetLite, RawNet2, and AASIST-LA) to provide context against more modern anti-spoofing systems.

5.1 Headline metrics

Table 1: DEV threshold (minimum EER) applied to EVAL.

System	Dev EER (%)	Eval EER (%)	Eval ACC (%)
LFCC + GMM	0.57	9.57	78.49
MFCC + GMM	9.96	12.51	84.12
LFCC + CNN	0.03	12.78	63.67
MFCC + CNN	14.64	20.30	76.36
RawNetLite	34.53	22.95	78.22
RawNet2	79.04	68.69	32.83
AASIST_LA	66.56	64.83	35.60

As per Table 1, LFCC + GMM attains the best EVAL EER (9.57%); MFCC + GMM yields the highest EVAL accuracy (84.12%). The CNN with LFCC nearly saturates DEV (0.03% EER) but degrades on EVAL (12.78% EER), suggesting some overfitting or distribution sensitivity. MFCC + CNN is the weakest overall (20.30% EVAL EER) from the baseline.

Among the pretrained models, RawNetLite achieves 22.95% EVAL EER with 78.22% accuracy, placing it close to the baseline CNNs. RawNet2 performs worse, with 68.69% EVAL EER and 32.83% accuracy. AASIST_LA reaches 64.83% EVAL EER and 35.60% accuracy. These results show that the baseline systems outperform the pretrained models under this evaluation setup.

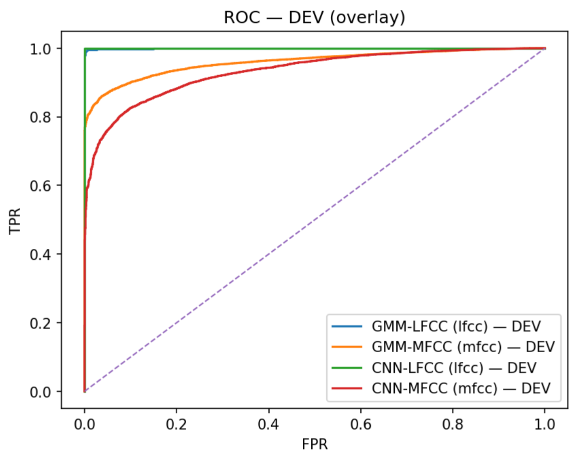
Compared to the baselines, all pretrained models perform worse under the ASVspoof 2019 LA protocol, despite being more complex; consistent with the fact that RawNetLite, RawNet2 and AASIST-LA were trained on different front-ends and different attack distributions (ASVspoof 2021 LA/DF), meaning their learned representations and calibration do not transfer cleanly to LA-2019. Notably, RawNetLite performs similarly to the CNNs, while RawNet2 and AASIST-LA collapse almost completely, illustrating the importance of matched preprocessing and domain alignment when using large waveform-based systems. This contrast highlights the robustness of simple LFCC/MFCC + GMM models when training and evaluation conditions are tightly controlled.

5.2 ROC curves

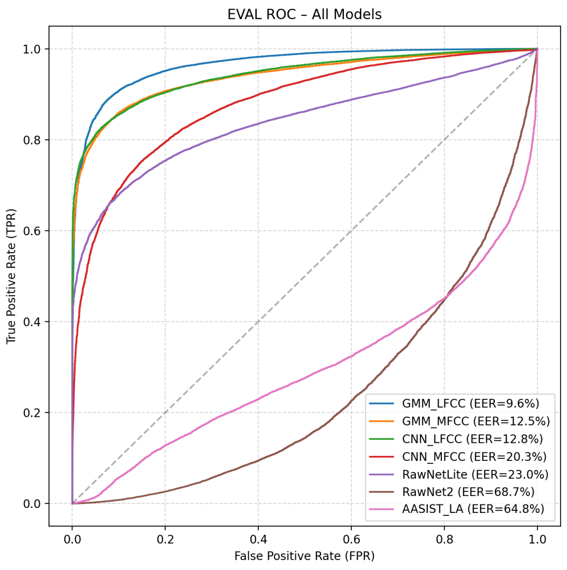
Figure 1 contrasts the DEV (Fig. 1a) and EVAL (Fig. 1b) ROC overlays.

On **DEV** (Fig. 1a), *CNN–LFCC* hugs the upper left corner; this is consistent with its near zero DEV EER; both GMM baselines trace smooth curves; *CNN–MFCC* sits below the others across most of the FPR range. The separation is largest at very low FPR; systems differ most in the strict false alarm regime.

On **EVAL** (Fig. 1b), performance compresses and curves spread less. *GMM–LFCC* dominates the low FPR region (FPR $\leq 1\%$); *GMM–MFCC* is close; *CNN–LFCC* drops below the GMMs; *CNN–MFCC* is consistently weakest. Relative to DEV, both CNN curves shift downward; this suggests sensitivity to distributional shift and calibration. In contrast, the GMMs retain shape and ranking, especially in the deployment relevant band FPR $\leq 1\%$.



(a) DEV split (overlay).



(b) EVAL split (overlay).

Figure 1: ROC curves on DEV and EVAL splits (overlays).

Overall, the ROC profiles indicate: (i) LFCC features favour better low FPR behaviour with GMMs; (ii) the CNN gains on DEV do not fully carry over to EVAL; (iii) model choice affects robustness more than feature choice when moving from DEV to EVAL.

The pretrained systems also appear in the combined EVAL ROC plots with RawNetLite tracking near the CNNs but remaining below the GMMs across the FPR range. RawNet2 and AASIST_LA sit lower on the ROC plane, with noticeably reduced true positive rates at all operating points. Their lower position reflects differences in their original training setup and preprocessing, which leads to reduced separation between bona fide and spoof scores under the LA evaluation protocol.

5.3 Per attack analysis on EVAL

According to Table 2, the following observations can be made:

1. *Easier attacks:* A07; A08; A09; and A16 are largely solved by LFCC + GMM (EER $\leq$ 0.46%); LFCC + CNN is also very strong on A07; A09; A16; A19.
2. *Challenging attacks:* A17 and A19 are consistently difficult for GMMs (especially MFCC + GMM on A19); the CNN struggles on A12; A13; and A18.
3. *Feature effect:* LFCC tends to help at the tails and at low FPR; MFCC sometimes yields higher overall accuracy via different score distributions.

5.4 Takeaways

- **Best EVAL EER** is achieved by **LFCC + GMM** (9.57%).
- **Best EVAL ACC** is **MFCC + GMM** (84.12%); this comes at a higher EER than LFCC + GMM.
- **CNN generalisation** is more variable; excellent DEV fit (LFCC) but weaker EVAL; this highlights the value of the simple generative baseline for robustness on this corpus.

Table 2: Per attack EVAL EER (%).

Attack	LFCC+GMM	MFCC+GMM	LFCC+CNN	MFCC+CNN
A07	0.33	6.49	0.01	13.41
A08	0.29	2.41	0.02	22.34
A09	0.06	0.02	0.02	6.49
A10	12.70	9.18	14.60	14.47
A11	2.28	1.57	2.36	9.62
A12	1.29	0.69	17.68	14.59
A13	6.08	5.82	28.47	23.47
A14	8.21	3.77	0.85	26.76
A15	16.08	6.29	1.93	18.05
A16	0.46	9.50	0.01	17.31
A17	24.04	34.65	26.87	33.33
A18	10.94	15.10	22.54	26.01
A19	5.81	36.08	0.05	22.53

- **Attack sensitivity** remains; A17 and A19 are the main failure modes across systems and are prime targets for future improvement.

6 Discussion

This section interprets the empirical findings; examines error patterns by attack family; reflects on design choices and efficiency; and outlines limitations and future directions.

6.1 Result analysis

Operating point and transfer A single DEV learned threshold applied to EVAL yields stable performance for the GMMs; the compact CNN shows sharper separation on DEV yet weaker transfer on EVAL. This points to sensitivity to distributional differences and to the fixed crop policy rather than a deficiency in the decision rule.

LFCC versus MFCC LFCC improves behaviour in the low FPR band; MFCC sometimes preserves accuracy at the same threshold. The likely link is retention of high frequency detail under linear spacing; artefacts that benefit LFCC are not uniformly present across attacks; this explains mixed accuracy outcomes without asserting a universal ordering.

Generative versus discriminative Frame level density modelling with fixed cepstra transfers reliably across splits; the CNN benefits from representation learning but requires stronger regularisation and crop strategy to maintain its margins when the score distribution shifts on EVAL.

Pretrained waveform and graph models The pretrained models provide an informative contrast to the baselines. Although they are architecturally more powerful, their performance collapses under LA-2019 evaluation (Table 1), with RawNet2 and AASIST-LA yielding EERs above 60%. This behaviour reflects a strong dependence on their original front-ends (pre-emphasis, sinc filters, normalisation) and the specific 2021 attack distributions they were trained on. When these conditions are not matched, the decision boundaries do not transfer. In contrast, the LFCC/MFCC + GMM models, despite their simplicity, remain stable because they rely on fixed, interpretable spectral statistics rather than learned waveform patterns.

6.2 Error analysis by attack

Patterns rather than IDs Attacks with stronger high frequency artefacts are well handled by LFCC based systems; attacks with weak or temporally local artefacts remain difficult for all four models. This suggests feature–attack mismatch more than a single classifier failure; lightweight score fusion is a natural remedy.

Operating region The advantage of LFCC + GMM concentrates at low FPR; this aligns with deployment needs where false alarms are costly. Where curves cross at higher FPR, MFCC + GMM can recover accuracy; this reflects different score distributions rather than a universal feature preference.

6.3 Design choices, efficiency, and reproducibility

Controls Global standardisation is fitted on training only and reused on DEV and EVAL; there is no per utterance renormalisation; there is no refitting after selecting the DEV threshold. These controls reduce leakage and make results reproducible.

Efficiency The GMM achieves competitive EVAL performance with a small parameter budget and modest compute; the compact CNN incurs additional convolutional cost without surpassing the GMM on EVAL under the present regularisation. The pretrained models require substantially more compute and specialised hardware, yet still perform worse than the efficient baselines in this evaluation.

6.4 Limitations and threats to validity

- **Dataset scope:** Results reflect LA conditions; transfer to other channels or codecs is unmeasured.
- **Metric scope:** EER is primary; calibration and cost sensitive evaluation are outside scope.
- **Uncertainty:** Confidence intervals and paired tests are not yet reported; these will be added.
- **Sensitivity:** CNN performance may depend on crop length; stride; and augmentation.
- **Pretrained models:** RawNetLite, RawNet2 and AASIST–LA are used with simplified front–ends; their results may reflect preprocessing mismatch rather than model capacity.

6.5 Implications and next steps

- **Front ends:** Test constant Q cepstra and simple context stacking; consider phase derived cues.
- **Regularisation:** Mixup on time and quefrency; masking; mild shifts; confidence penalty.
- **Fusion:** Logistic fusion of GMM and CNN scores learned on DEV to reduce per attack variance.
- **Robustness:** Mild compression; additive noise; and resampling stress tests at the fixed threshold.
- **Statistical reporting:** Add confidence intervals; paired tests; and the efficiency table as standard.
- **Pretrained models:** Reproduce full RawNet2 and AASIST front–ends or fine–tune them on LA to enable fairer comparison with the cepstral baselines.

7 Conclusion

This work implemented a reproducible baseline for deepfake speech detection on the ASVspoof 2019 Logical Access corpus using two compact cepstral front ends (LFCC; MFCC) and two lightweight classifiers (GMM; CNN). The pipeline was designed for transparency, reproducibility, and CPU friendliness while adhering to official challenge protocols.

Across all configurations, **LFCC + GMM** achieved the best evaluation EER (9.57%) and consistent low-FPR behaviour; **MFCC + GMM** achieved slightly higher accuracy at the same threshold. The CNNs exhibited stronger discrimination on the development split but reduced generalisation to evaluation, indicating that model regularisation and crop strategy play a larger role than feature choice under limited data.

These results confirm that traditional cepstral front ends paired with simple generative classifiers remain competitive and robust against modern deepfake speech attacks when implemented carefully. They also highlight that apparent improvements from deeper architectures may not persist across dataset partitions unless explicitly regularised and cross-validated.

The pretrained RawNetLite, RawNet2, and AASIST–LA models provided an additional reference point by showing how larger waveform and graph-based systems behave under the same fixed DEV$\rightarrow$EVAL protocol. Their results offer useful context for interpreting the baselines and demonstrate how different modelling approaches respond when evaluated within a common, reproducible framework.

Future work could include more detailed preprocessing for the pretrained models, or fine-tuning them directly on the ASVspoof 2019 LA data. This would allow a fairer comparison with the baselines. Other possible extensions include data augmentation, alternative feature representations such as CQCC, or combining cepstral and waveform features.

In summary, the project answers its research question affirmatively: *LFCC + GMM can reach competitive EER within the 10–15% target while remaining reproducible and resource-efficient.* The resulting system serves as a transparent and extensible baseline for subsequent countermeasure research and teaching.

References

[1] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans, and H. Delgado, “Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection,” 09 2021, pp. 47–54.

[2] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, L. Juvela, P. Alku, Y.-H. Peng, H.-T. Hwang, Y. Tsao, H.-M. Wang, S. L. Maguer, M. Becker, F. Henderson, R. Clark, Y. Zhang, Q. Wang, Y. Jia, K. Onuma, K. Mushika, T. Kaneda, Y. Jiang, L.-J. Liu, Y.-C. Wu, W.-C. Huang, T. Toda, K. Tanaka, H. Kameoka, I. Steiner, D. Matrouf, J.-F. Bonastre, A. Govender, S. Ronanki, J.-X. Zhang, and Z.-H. Ling, “Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech,” *Computer Speech & Language*, vol. 64, p. 101114, 2020. [Online]. Available: https://doi.org/10.1016/j.csl.2020.101114

[3] J. Yamagishi, M. Todisco, M. Sahidullah, H. Delgado, X. Wang, N. Evans, T. Kinnunen, K. A. Lee, V. Vestman, and A. Nautsch, “Asvspoof 2019: Automatic speaker verification spoofing and countermeasures challenge evaluation plan,” ASVspoof Consortium, Tech. Rep., Jan. 2019. [Online]. Available: http://www.asvspoof.org/

[4] M. Sahidullah, T. Kinnunen, and C. Hanilçi, “A comparison of features for synthetic speech detection,” 09 2015.

[5] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, and M. Sahidullah, “Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge,” 09 2015.

[6] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, “The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection,” 08 2017.

[7] T. Kinnunen, N. Evans, J. Yamagishi, K. A. Lee, M. Sahidullah, M. Todisco, and H. Delgado, “Asvspoof 2017: Automatic speaker verification spoofing and countermeasures challenge evaluation plan *,” 09 2018.

[8] “Asvspoof 2019 challenge website,” https://www.asvspoof.org/index2019.html.

[9] “Asvspoof 2021 challenge website,” https://www.asvspoof.org/index2021.html.

[10] X. Zhou, D. Garcia-Romero, R. Duraiswami, C. Espy-Wilson, and S. Shamma, “Linear versus mel frequency cepstral coefficients for speaker recognition,” 12 2011.

[11] M. Todisco, H. Delgado, and N. Evans, “Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification,” *Computer Speech Language*, vol. 45, 02 2017.

[12] ASVspoof Challenge organizers, “ASVspoof 2021 Baseline Systems,” https://github.com/asvspoof-challenge/2021, 2021, gitHub repository.

[13] G. Lavrentyeva, S. Novoselov, T. Andzhukaev, M. Volkova, A. Gorlanov, and A. Kozlov, “Stc antispoofing systems for the asvspoof2019 challenge,” pp. 1033–1037, 09 2019.

[14] J.-W. Jung, H.-S. Heo, H. Tak, H.-J. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, “Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks,” 05 2022, pp. 6367–6371.

[15] A. Gomez-Alanis, A. Peinado, J. Gonzalez Lopez, and A. Gomez, “A light convolutional gru-rnn deep feature extractor for asv spoofing detection,” 09 2019, pp. 1068–1072.

[16] ——, “A gated recurrent convolutional neural network for robust spoofing detection,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, pp. 1985–1999, 08 2019.

[17] T. Kinnunen, K. A. Lee, H. Delgado, N. Evans, M. Todisco, M. Sahidullah, J. Yamagishi, and D. A. Reynolds, “t-dcf: a detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification,” 2019. [Online]. Available: https://arxiv.org/abs/1804.09618