Audio Engineering Society

Conference Paper 3

Presented at the AES 8th International Conference on Audio Forensics
2024 June 27–29, Denver, Colorado, USA

This paper was peer-reviewed as a complete manuscript for presentation at this conference. This paper is available in the AES E-Library (http://www.aes.org/e-lib), all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Calibrating neural networks for synthetic speech detection: A likelihood-ratio-based approach

Luca Cuccovillo, Patrick Aichroth, and Thomas Köllmer

Fraunhofer Institute for Digital Media Technology IDMT

Correspondence should be addressed to Luca Cuccovillo (luca.cuccovillo@idmt.fraunhofer.de)

ABSTRACT

In this paper, we introduce a calibration procedure designed to convert the uncalibrated output scores of neural networks for synthetic speech detection into calibrated and interpretable likelihood ratios. This procedure is based on the assumption that the networks subject to calibration are deterministic and have undergone training until they reached convergence. Provided these conditions are satisfied, it is then possible to transform their output values into likelihood ratios using a minimal set of validation and calibration data, eliminating the need for retraining the models. We successfully tested the entire workflow on a state-of-the-art network example, demonstrating not only its effectiveness in calibration but also its ability to enhance fault tolerance against inadequate inputs.

1 Introduction

Synthetic content, which reached unprecedented quality after the advent of artificial intelligence (AI), has been identified by a Europol report as a critical threat that could become a staple tool for organized crime [1].

According to the report, the abuse of content synthesis could enable various criminal activities, including targeting individuals for harassment or humiliation, committing extortion and fraud, creating false online identities, engaging in online child sexual exploitation, manipulating electronic evidence, distributing disinformation, fomenting social unrest and political polarization, and amplifying the narratives of extremist or terrorist groups.

The current concerns about the potential misuse of synthetic speech were partially anticipated by the research community, and in particular by the organizers of the ISCA bi-annual anti-spoofing challenge for automatic speaker verification (ASVspoof), who, since 2019, have included synthetic speech samples in their dataset with the aim of investigating the potential of AI-based speaker impersonation to fool automatic speaker verification systems. The evaluation highlighted the urgent need for robust and effective speech detection systems, with the result that since ASVspoof 2019 research papers on synthetic speech detection became progressively more numerous, and improved in accuracy and complexity [2].

However, the rush to develop effective speech detection systems has encountered several challenges: the existing open datasets do not meet the technical requirements for producing market-ready detection systems; neural networks designed for detection suffer from lack of interpretability and struggle with out-of-distribution input data; there is limited collaboration between in-

dustrial and research institutions [3].

Furthermore, due to the competitive nature of ASVspoof and similar challenges, existing solutions are evaluated on metrics which are threshold-independent, such as the equal error rate (EER), the total area under the curve (AUC) of the receiver operating characteristic curves (ROC curves), or the tandem detection cost function (t-DCF) proposed by the ASVspoof organizers in [4], to assess both synthesis detection and anti-spoofing capabilities effectively.

While these threshold-independent metrics are effective for challenges and serve the need for precise and reproducible performance evaluation, the systems themselves require a detection threshold to be set by the end-user. However, the selection of this threshold, which largely influences the final outcome, remains subjective and arbitrary – making it unsuitable for reproducible and objective forensic examinations.

In this paper, we propose a novel procedure aimed at overcoming this limitation by enabling the calibration and precise interpretation of neural networks output scores for speech synthesis detection using likelihood ratios (LRs).

Following this procedure offers several significant benefits: LR values enable a straightforward interpretation, and can be delivered to the court both numerically and in their verbal equivalent; the files used for LR calibration can be fully documented and inspected, unlike the data used for training the neural network; LR calibration avoids the need for expensive (and sometimes unfeasible) retraining of the underlying AI-based detection algorithm, and it can adequately reflect the quality and characteristics of the evidence under analysis.

The rest of the paper is organized as follows: Section 2 introduces the fundamental assumptions behind our calibration procedure, which is then elaborated on in Section 3; Section 4 outlines the evaluation setup, with results discussed in Section 5; Section 6 concludes with a summary and perspectives for future research.

2 Fundamental Assumptions

The calibration procedure is based on two simple yet fundamental assumptions that delineate the scope within which it can be effectively applied.

Assumption 1 *The detection system $D : \mathbb{R}^L \to \mathbb{R}^M$ is a deterministic function that maps an input speech signal of finite length $x \in \mathbb{R}^L$ to a finite vector of continuous output values $y \in \mathbb{R}^M$.*

Assumption 1 implies that the model D is deterministic, i.e., it is entirely defined by a set of parameters Θ_D, and that it transforms the input into a vector with a finite number of continuous elements.

In AI-based systems, a prevalent choice for D is to output a single scalar, bound within two limit values, e.g., through a sigmoid activation function

$$D(x \mid \Theta_\mathrm{D}) = \frac{1}{1 + e^{-z(x|\Theta_\mathrm{D})}}, \tag{1a}$$

or by means of a hyperbolic tangent activation function

$$D(x \mid \Theta_\mathrm{D}) = \frac{e^{2z(x|\Theta_\mathrm{D})} - 1}{e^{2z(x|\Theta_\mathrm{D})} + 1}, \tag{1b}$$

where $z(\cdot \mid \Theta_\mathrm{D})$ is a common notation to denote the output of the entire network up to the last activation, often referred to as "logit output."

A less common choice is to have the network output a 2-dimensional vector with a softmax activation function, where the i-th element is defined by

$$D(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}, \tag{1c}$$

with z_i and z_j being the i-th and j-th elements of the logit output $z(\cdot \mid \Theta_\mathrm{D}) \in \mathbb{R}^M$.

According to the assumption, it is also possible – though, to our knowledge, not yet investigated in the literature – to view the network as a mere feature extraction tool with an output embedding vector in $N > 2$ dimensions, the distribution of which can be used to distinguish synthetic speech from natural one:

$$D(x \mid \Theta_\mathrm{D}) = z(x \mid \Theta_\mathrm{D}), \tag{1d}$$

where, again, Θ_D represents the model parameters, and the individual dimensions may or may not indicate specific characteristics of the input speech[1].

Assumption 2 *The detection system $D : \mathbb{R}^L \to \mathbb{R}^M$ was trained so that its output values y are similar for training input with the same class label c, and dissimilar otherwise.*

[1] e.g., emotion embeddings, i-vectors, F0 trajectory...

Assumption 2 implies that the detection model was trained correctly – e.g., by leveraging appropriate loss functions such as binary cross-entropy, sparse categorical cross-entropy, or triplet loss – and is thus capable of distinguishing between known test examples from the training and validation distributions.

3 Proposed Method

3.1 Output Score Distribution

Due to the binary nature of the synthetic speech detection problem, the vast majority of state-of-the-art networks for synthetic speech detection [5, 6, 7, 8] train the synthesis detection network using binary cross-entropy loss, designed to measure the dissimilarity between the predicted output $\hat{y} \in [0,1]$ and the the target output values $y \in \{0,1\}$:

$$L(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i), \quad (2)$$

where N is the number of samples in the dataset, and $y_i, \hat{y}_i$ denote the true label and predicted value, respectively. Since the value $y = 1$ often denotes an alarm, it commonly interpreted that the output score reflects the probability of the input signal x being synthetic.

The distribution of the output logit scores of such neural networks is depicted for reference in Figure 1, in which the y-axis represents the scores, and the x-axis represents the index of the test examples. The data distribute as expected considering the adoption of the binary cross-entropy loss, showing extreme values symmetric with respect to the origin $y = 0$, indicating that

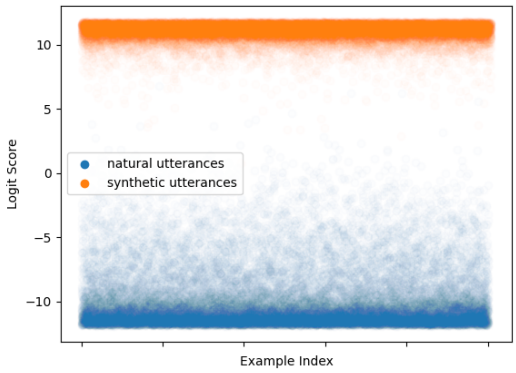


Fig. 1: Example distribution of the logit outputs of synthetic speech detection systems

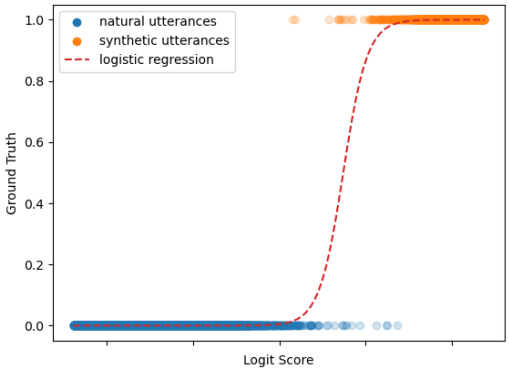


Fig. 2: Logistic Regression Model

the output values tend to be biased towards the two extremes, with a notably sparse density of output values in the center region.

The output values for the two classes do not follow a Gaussian distribution, nor are symmetric with respect to a class-specific mean value. Indeed, due to the interaction between the activation function and the binary cross-entropy loss, they are distributed by design according to a logistic regression model – as illustrated in Figure 2, in which the x-axis represents the score of the test examples, and the y-axis the associated ground truth, with 1 indicating synthetic speech.

3.2 LR Calibration Procedure

Let us consider the two following complementary hypotheses:

H_1 : The input speech is synthetic

H_0 : The input speech is *not* synthetic, but natural

The LR calibration procedure consists of two consecutive steps:

1. Modeling a probability distribution $P_{\text{score}}(H_1 \mid \hat{y})$ to the output scores $\hat{y}$ computed by the network on a *validation* dataset x_{score}.

2. Fitting the two distributions $P_{\text{calib}}(\hat{p}_{\text{calib}} \mid H_1)$ and $P_{\text{calib}}(\hat{p}_{\text{calib}} \mid H_0)$ to the output probabilities $\hat{p}$ computed by the previous model on a *calibration* dataset x_{calib}, under the two hypotheses.

After both steps are completed, given an input file x, the corresponding calibrated LR would be equal to

$$LR(x) = \frac{\mathcal{L}(H_1 \mid x)}{\mathcal{L}(H_0 \mid x)} = \frac{P_{\text{calib}}(P_{\text{score}}(H_1 \mid \hat{y}) \mid H_1)}{P_{\text{calib}}(P_{\text{score}}(H_1 \mid \hat{y}) \mid H_0)}, \quad (3)$$

where

$$\hat{y} = D(x \mid \Theta_{\text{D}}), \quad (4)$$

with Θ_{D} the neural network parameters.

Fitting the output score distribution

The first step of the LR calibration involves modeling a probability distribution $P_{\text{score}}(H_1 \mid \hat{y})$ to the output scores $\hat{y}$ computed by the network on a *validation* dataset x_{score}.

Based on observations in the previous section, we propose using a logistic regression model for this purpose, defined for a single dimension by:

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1)}}, \quad (5)$$

with β_0 and β_1 the logistic regression parameters to be estimated through standard techniques, e.g., the Newton's method [9] or stochastic gradient descent [10].

Logistic regression not only perfectly matches the output score distribution, but also allows for an expansion to M explanatory variables, defined as a function $p : \mathbb{R}^M \rightarrow [1, 0]$ of the form:

$$p(\mathbf{x} \mid \beta_{\text{score}}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_M x_M)}}, \quad (6)$$

where $\beta_{\text{score}} = \{\beta_0, \beta_1, \ldots, \beta_M\}$ comprises the set of $M + 1$ model parameters.

Therefore, logistic regression enables the conversion of any network output vector to a single scalar, indicating the probability of the input utterance being synthetic or natural.

If the assumptions in Section 2 are met, we can establish the regression parameters with a validation set and assert:

$$P_{\text{score}}(H_1 \mid \hat{y}) = p(\hat{y} \mid \beta_{\text{score}}), \quad (7)$$

implying that the output $\hat{y}$ of the synthesis detection network can be mapped to an uncalibrated probability.

Fitting the output score distributions for LR

The second and final step of LR calibration consists of fitting the two distributions $P_{\text{calib}}(\hat{p}_{\text{calib}} \mid H_1)$ and $P_{\text{calib}}(\hat{p}_{\text{calib}} \mid H_0)$ to the output probabilities $\hat{p}$ computed by the aforementioned logistic regression model on a *calibration* dataset x_{calib}, under both hypotheses.

The distributions can be identified by collecting representative examples of the two hypothesis. This is achieved by:

1. Collecting a calibration set of synthetic audio files:

$$X_{\text{calib},H_1} = \{x_{H_1}^{(i)}\}$$

2. Calculating the synthesis probabilities for each sample:

$$\hat{P}_{\text{calib},H_1} = \left\{P_{\text{score}}(H_1 \mid x_{H_1}^{(i)})\right\}$$

3. Fitting the output set of probabilities:

$$P_{\text{calib}}(\hat{p} \mid H_1) = \text{fit}\left(\hat{P}_{\text{calib},H_1}\right)$$

And, similarly for natural audio files, by:

1. Collecting a calibration set of natural audio files

$$X_{\text{calib},H_0} = \{x_{H_0}^{(i)}\}$$

2. Calculating the synthesis probabilities for each sample

$$\hat{P}_{\text{calib},H_0} = \left\{P_{\text{score}}(H_1 \mid x_{H_0}^{(i)})\right\}$$

3. Fitting the output set of probabilities:

$$P_{\text{calib}}(\hat{p} \mid H_0) = \text{fit}\left(\hat{P}_{\text{calib},H_0}\right).$$

Given the variability in network training and data selection, we cannot predetermine the exact form of neither $P_{\text{calib}}(\hat{p} \mid H_0)$ nor $P_{\text{calib}}(\hat{p} \mid H_1)$. Therefore, we suggest relying on kernel density estimation (KDE), a non-parametric method suited for estimating probability density functions (PDFs) without presuming a specific distribution [11].

The basic idea behind KDE is to place a kernel – a smooth, symmetric, and usually bell-shaped function

– at each data point, and then sum up these kernels to create a smooth estimate of the PDF.

The KDE at a point X is given by:

$$KDE(x \mid \{x_i\}) = \frac{1}{hN} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right), \tag{8}$$

with N the number of reference data points x_i, $K(\cdot)$ the kernel function – typically Gaussian – and h the bandwidth, a smoothing parameter that controls the kernel width.

After both distributions have been fit with KDE, i.e.,

$$P_{\text{calib}}(\hat{p} \mid H_1) = KDE\left(\hat{p} \mid \hat{P}_{\text{calib},H_1}\right) \tag{9}$$

$$P_{\text{calib}}(\hat{p} \mid H_0) = KDE\left(\hat{p} \mid \hat{P}_{\text{calib},H_0}\right) \tag{10}$$

the output $\hat{y}$ of the synthesis detection network can be mapped to a calibrated LR as follows:

$$LR(\hat{y}) = \frac{KDE\left(p(\hat{y} \mid \beta_{\text{score}}) \mid \hat{P}_{\text{calib},H_1}\right)}{KDE\left(p(\hat{y} \mid \beta_{\text{score}}) \mid \hat{P}_{\text{calib},H_0}\right)} \tag{11}$$

where

$$\hat{y} = D(x \mid \Theta_{\text{D}}), \tag{12}$$

with Θ_{D} the neural network parameters.

4 Experimental Setup

4.1 Dataset

We conducted all the experiments on the ASVspoof2019 speech dataset [12], specifically designed for the ASVspoof2019 challenge. This dataset includes speech utterances created for competitors to analyze, facilitating the dual tasks of speech spoofing detection and automatic speaker verification.

In particular, ASVspoof includes a Logical Access (LA) set of files containing both natural and synthetic voice utterances, and is divided into three partitions called *train*, *dev* and *eval*. All partitions include natural speech signals along with synthetic speech samples generated through various methods.

The *train* and *dev* utilize the same set of six synthesis algorithms (A01, A02, ..., A06), while the *eval* partition consists of samples generated by thirteen distinct techniques (A07, A08, ..., A19). This configuration simulates an open scenario where attacks are executed using unknown algorithms trained on unknown voices, since only two algorithms in the eval partition are included in the training and dev partitions.

We pre-processed all audio files in the set to normalize their volume according to the EBU recommendation regarding audio signal loudness [13], using a sampling rate of 16,000 Hz. Furthermore, we applied the MarbleNet model for voice activity detection [14] to remove any leading and trailing silence from the files, and therefore to prevent an uneven distribution of noise from biasing the results [15].

We merged the train and devel partition of ASVspoof2019 LA into a training subset in two versions, in the following referred to as T_{pcm} and T_{lossy}. The first version consisted only of PCM files; the second was generated by keeping one-fifth of the content of T_{pcm} unaltered, and by compressing the remaining four-fifths using AAC with bitrates equal to 128, 64, 32, and 16 kbps, with each bitrate representing 20% of the dataset.

The eval partition of ASVspoof2019 was further divided into three subsets: 10% for validation (X_{score}), 10% for calibration (X_{calib}), and 80% for testing (X_{test}). All segments of the validation, calibration, and test subsets were encoded with AAC using the bitrates specified above, to ensure consistency across the evaluation process. To denote a specific combination of split and encoding, we use both indicators – e.g., $X_{(\text{calib,aac-32})}$ indicates calibration content encoded with AAC using 32 kbps.

The encoding to prepare the training, validation, calibration, and testing subsets relied on the default AAC encoder bundled with ffmpeg 4.2.7, as part of the libavcodec 58.54.100 library.

4.2 Application scenario

The evaluation assumed a hypothetical application scenario, in which a pre-trained model for synthesis detection is validated, and then calibrated and tested to evaluate its robustness against lossy encoding.

The validation data x_{score} used to learn how to translate the model output into the uncalibrated probability score $P_{\text{score}}(H_1 \mid \hat{y})$ will therefore be fixed:

$$x_{\text{score}} = X_{(\text{score,pcm})}. \tag{13}$$

The calibration data x_{calib} used to fit the LR distributions $P_{\text{calib}}(\hat{p} \mid H_1)$ and $P_{\text{calib}}(\hat{p} \mid H_0)$ and the test data x_{test} will instead be varying across encoding conditions:

$$x_{\text{calib}} \in \{X_{(\text{calib,enc})}\}, \tag{14}$$

$$x_{\text{test}} \in \{X_{(\text{test,enc})}\}, \tag{15}$$

with

$$\text{enc} \in \{\text{pcm}, \text{aac-128}, \text{aac-64}, \text{aac-32}, \text{aac-16}\}. \tag{16}$$

The case enc = pcm corresponds to the most favorable working conditions: The calibrated system will evaluate lossless content created by unknown synthesis algorithm for unknown voices.

The case enc = aac-* represents a more critical scenario, for which we aim to understand whether the network can operate with those encoding conditions or not, and to determine if after calibration the performance for pre-defined LR thresholds are stable across different encoding conditions or not.

4.3 Synthesis detection network

The LR calibration procedure was evaluated by relying on SFAT-Net-2 [16] for the detection process.

SFAT-Net-2 is a synthetic speech detection network based on a feature bottleneck that encodes the magnitude and phase of the input speech to perform three joint tasks: reconstructing the magnitude of the input signal, predicting the fundamental frequency (F0) of the input speech, and discerning if the input utterance is synthetic or natural.

The network is making use of sequence-to-sequence transformers and of their attention mechanism to detect non-local relations in the input space. Moreover, thanks to the feature bottleneck mentioned above, the architecture is forced to model accurately the energy and phase of the input speech formants, thereby basing the detection not on the sheer distribution of the input data, but on the behavior of the input speech formants – i.e., with an explainable detection mechanism.

The network was trained in two versions, one using the T_{pcm} dataset and one using the T_{lossy} dataset. The training utilized input audio segments of 2.064 seconds, converted into a short-time Fourier transform (STFT) with 128 frames and 256 frequency bins, obtained by extracting windows of 32 msec every 16 msec.

To address the class imbalance between synthetic and natural examples inherent in ASVspoof 2019 LA, we employed oversampling of the natural recordings, and ensured the uniqueness of training segments by randomizing the starting offset applied to the file before extracting each interval of interest.

The training procedure spanned 100 iterations, using Adam optimization coupled with cosine annealing. Cosine annealing was setup in order complete a full cycle every 2 epochs.We used mini-batches of 64 elements, and an initial learning rate of 1e-4. Due to space limitations, other parameters, specific to the model's configuration rather than the training methodology, are omitted but detailed in [16].

4.4 Evaluation metrics

The average performance of the calibrated systems will be assessed using the log-likelihood-ratio cost (C_{llr}) and discriminating power ($C_{\text{llr}}{}^{\text{min}}$) metrics, commonly employed to evaluate speaker verification systems that rely on LRs.

Let LR_{H_1} denote the set of LRs computed for synthetic test trials in x_{test}, and LR_{H_0} the set of LRs computed for natural test trials. C_{llr} is an accuracy metric defined by

$$C_{\text{llr}} = \frac{1}{2}\left(\frac{1}{|LR_{H_1}|}\sum_{l_i \in LR_{H_1}} \log_2 \frac{l_i+1}{l_i} + \ldots + \frac{1}{|LR_{H_0}|}\sum_{l_j \in LR_{H_0}} \log_2(1+l_j)\right) \tag{17}$$

measuring the cost of erroneous decisions – false alarms and false rejections – made on the test content. An ideal value of 0 would correspond to a perfect system with no errors [17].

Let y_i be the ground truth label equal to 1 or 0 of the i-th test sample, and l_i its corresponding likelihood ratio. Each pair in the test set is used to fit a isotonic regression function $\hat{y} = f_{\text{min}}(x)$, which solves the following problem:

$$\min \sum_i w_i (y_i - \hat{y}_i)^2, \tag{18}$$

subject to $\hat{y}_i \leq \hat{y}_j$ whenever $x_i \leq x_j$, with the weights w_i being strictly positive.

Then, $C_{\text{llr}}{}^{\text{min}}$ can be determined by computing the C_{llr} of the transformed likelihood ratios

$$l_i{}^{\text{min}} = f_{\text{min}}\left(\frac{l_i}{1+l_i}\right), \tag{19}$$

Table 1: Performance of SFAT-Net-2, if trained only on lossless PCM content (T_{pcm})

τ_{LLR}	PCM-16		AAC-128		AAC-64		AAC-32		AAC-16	
	precision	recall	precision	recall	precision	recall	precision	recall	precision	recall
0	0.8601	0.8054	0.7896	0.6722	0.7921	0.6646	0.6943	0.6247	0.6523	0.6523
1	0.9942	0.5011	N/A	0	N/A	0	N/A	0	N/A	0
2	0.9988	0.4302	N/A	0	N/A	0	N/A	0	N/A	0

(a) punctual metrics for fixed threshold values

Metric	Encoding				
	PCM-16	AAC-128	AAC-64	AAC-32	AAC-16
C_{llr}	0.552	0.751	0.751	0.871	0.904
$C_{\text{llr}}^{\text{min}}$	0.520	0.736	0.734	0.859	0.891

(b) global metrics based on likelihood-ratio costs

by replacing the l_i terms in eq. (17) with their equivalent l_i^{min}. $C_{\text{llr}}^{\text{min}}$ denotes the discriminatory capability for the two classes, representing the C_{llr} for a perfectly calibrated system. An optimal value of 0 would indicate a flawless system [18].

Furthermore, we aim to assess the system performance across each lossy-encoding bitrate by applying a threshold τ_{LLR} to the LRs obtained by the calibrated system, and then computing precision and recall.

Given the following definitions of True Positive (TP) and True Negative (TN),

- TP: A synthetic test sample x_i s.t. $\log_{10} l_i \geq \tau_{\text{LLR}}$;
- TN: A natural test sample x_i s.t. $\log_{10} l_i < \tau_{\text{LLR}}$;

Precision (P) and Recall (R) can be computed according to:

$$P = \frac{TP}{TP+FP} \quad R = \frac{TP}{TP+FN}, \tag{20}$$

facilitating an understanding of specific operating conditions for different LLR thresholds.

5 Evaluation Results

5.1 Detection network trained on lossless recordings and tested on lossy content

For the first test scenario, SFAT-Net-2 was trained exclusively on the lossless training dataset T_{pcm}, and evaluated on lossy-encoded content. The corresponding performance measures are depicted in Table 1.

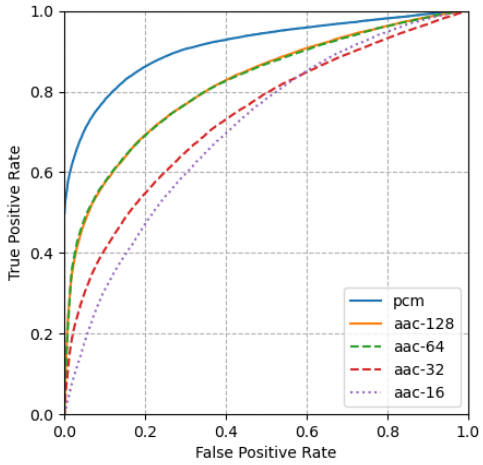


Fig. 3: ROC curves of the baseline detection system, trained on uncompressed PCM-16 content

From the precision and recall measures in Table 1, it is evident that the system struggles with the analysis of encoded content: While handling PCM input adequately, the model's effectiveness diminishes even at the minimal lossy encoding bitrate of 128 kbps, where LRs consistently fail to exceed 1, indicating weak support in the conclusions.

Similarly, both the log-likelihood-ratio cost C_{llr} and the discriminating power $C_{\text{llr}}^{\text{min}}$ measures increase abruptly and approach 1.0 as soon as a minimal amount of lossy-encoding is applied to the test content. This indicates a substantial overlap between the LR distribu-

Table 2: Performance of SFAT-Net-2, if also trained on AAC lossy-encoded content (T_{lossy})

τ_{LLR}	PCM-16		AAC-128		AAC-64		AAC-32		AAC-16	
	precision	recall	precision	recall	precision	recall	precision	recall	precision	recall
0	0.8627	0.7927	0.8530	0.8041	0.8554	0.8022	0.8417	0.7714	0.7710	0.6653
1	0.9892	0.5030	0.9876	0.5232	0.9943	0.4961	0.9902	0.4816	0.9594	0.2644
2	0.9978	0.4048	0.9973	0.4508	0.9990	0.4035	0.9974	0.4067	0.9955	0.0550

(a) punctual metrics for fixed threshold values

Metric	Encoding				
	PCM-16	AAC-128	AAC-64	AAC-32	AAC-16
C_{llr}	0.559	0.550	0.548	0.580	0.759
$C_{\text{llr}}^{\text{min}}$	0.535	0.528	0.565	0.565	0.757

(b) global metrics based on likelihood-ratio costs

tions obtained for the two hypotheses.

If we consider the network as a black box which cannot be retrained – e.g, because it was provided as a commercial product – the results imply that the network might need to be replaced, or supplemented with a passive encoding detection system to reject any input containing traces of lossy encoding [19].

This observation is further supported by the ROC curves shown in Figure 3: The model's performance decreases substantially for lossy-encoded input test content, leading to the poor results in Table 1.

5.2 Detection network trained partially on lossy recordings and tested on lossy content

In the second test scenario, we continued to assess SFAT-Net-2 on AAC lossy-encoded content, but adapted the training to include the T_{lossy} dataset, where a portion of the original content underwent separate encoding. The corresponding performance outcomes are summarized in Table 2.

The precision and recall in Table 2 demonstrate a significant improvement over the results for the previous test scenario. Once again, the model maintains high efficacy for PCM input, and seems able to cope with unknown input algorithms synthesizing unknown voices. However, it also exhibits robust performance across AAC-encoded test content up to 32 kbps, achieving high precision and moderate recall for LRs higher than 2 – i.e., with high/moderate confidence. A noticeable

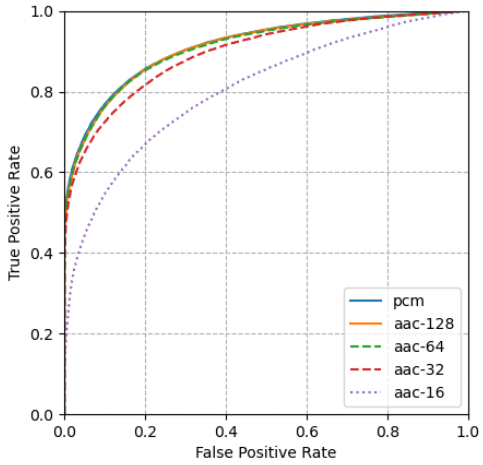


Fig. 4: ROC curves of the baseline detection system, trained on AAC lossy-encoded content

drop occurs at 16 kbps, however, where the system rarely identifies synthetic data with a LR greater than 2.

Moreover, both the C_{llr} and the $C_{\text{llr}}^{\text{min}}$ measures show only minimal variations across most encoding settings, with the notable exception of AAC-16, in which both values raise abruptly.

If we again treat the network as a black box which cannot be retrained, this evaluation suggests that the network can be used with substantial confidence even with AAC-encoded input content, as long as the bitrate is beyond 32 kbps. As for the first test scenario, this

weakness might be addressed by applying a passive encoding detection system that rejects any input audio with lower bitrates [19].

The outcome also aligns with the behavior evident from the corresponding ROC curves in Figure 4: The model performance are nearly unaffected by lossy encoding, with the only exception of AAC-16 which we also identified as an issue in Table 2.

6 Summary and Outlook

In this paper, we introduced a procedure to calibrate and interpret the output scores of neural networks for synthetic speech detection by leveraging likelihood ratios (LRs).

The proposed procedure is based on the assumptions that the network to be calibrated is deterministic and has been trained until convergence. It involves first fitting a logistic regression model to the model output scores obtained on a validation set, and then modeling the output probabilities obtained on a calibration set for both classes by means of kernel density estimation.

The resulting LRs offer a means to interpret – and threshold, if necessary – the model output in a clear and standardized way, such as by referring to the verbal equivalents of the numeric values as recommended by the European Network of Forensic Science Institutes (ENFSI) in their guideline for analysis reports [20]. This interpretability is crucial, as it addresses the limitations of the initial models, whose output range may vary significantly and unpredictably according to the technical characteristics of the input audio signals, such as encoding, background noise, and loudness.

We assessed the proposed calibration approach with two neural networks, one trained only on lossless content, and another trained with a mixture of lossless and lossy content, using both global C_{llr} and $C_{\text{llr}}{}^{\text{min}}$ measures designed for LR-based systems, and punctual precision and recall metrics to evaluate the network behavior for fixed threshold values.

The results are encouraging: The first network, challenged by lossy-encoded content, was swiftly flagged as problematic by the C_{llr} and $C_{\text{llr}}{}^{\text{min}}$ measures, showing an inability to output log-likelihood ratio (LLR) values above 1 for encoded content, indicating a lack of confidence in the produced output. The second network, instead, was capable of accurately analyzing lossy-encoded content, demonstrated improved C_{llr} and $C_{\text{llr}}{}^{\text{min}}$ measures, and produced LR values exceeding both 1 and 2 – thus providing *interpretable* output values which could be related to *interpretable* thresholds, in contrast to arbitrary output scores or uncalibrated probabilities.

Another positive outcome of the evaluation is that the C_{llr} and $C_{\text{llr}}{}^{\text{min}}$ measures were effective in identifying the limitations of both models – suggesting their potential to screen inputs with technical characteristics potentially leading to unreliable outputs. Thus, using LRs not only enhances interpretability of the model output, but it also improves fault tolerance against unsuitable inputs.

Looking ahead, we aim to explore this topic further, including assessing how the calibration responds to models other than logistic regression and with KDE being involved in the process. Moreover, we plan to compare several systems to establish benchmarks for C_{llr} and $C_{\text{llr}}{}^{\text{min}}$ that might serve as minimal requirements for adopting synthesis detection models.

7 Acknowledgments

The work for this paper was supported by the BMBF SpeechTrust+ project (grant no 13N16267), and by the BMBF news-polygraph project (grant no 03RU2U151D).

Additionally, we would like to thank Florian Braun (Fraunhofer IDMT) for his support in the initial experiments, based on the LIR Python Likelihood Ratio Library released by the Netherlands Forensic Institute.

References

[1] Europol, *Facing reality? Law enforcement and the challenge of deepfakes, an observatory report from the Europol Innovation Lab*, Publications Office of the European Union, 2022.

[2] Todisco, M. et al., "ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection," in *ISCA Annual Conference of the International Speech Communication Association (INTERPEECH)*, pp. 1008–1012, Graz, Austria, 2019.

[3] Cuccovillo, L., Papastergiopoulos, C., Vafeiadis, A., Yaroshchuk, A., Aichroth, P., Votis, K., and Tzovaras, D., "Open Challenges in Synthetic Speech Detection," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, Shanghai, China, 2022.

[4] Kinnunen, T. et al., "Tandem Assessment of Spoofing Countermeasures and Automatic Speaker Verification: Fundamentals," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28, pp. 2195–2210, 2020.

[5] Tak, H. et al., "End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection," in *ISCA Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pp. 1–8, Virtual, 2021.

[6] Attorresi, L. et al., "Combining Automatic Speaker Verification and Prosody Analysis for Synthetic Speech Detection," in *International Conference on Pattern Recognition (ICPR)*, pp. 247–263, Montréal Québec, 2022.

[7] Jun, J.-w. et al., "AASIST: Audio Anti-Spoofing Using Integrated Spectro-Temporal Graph Attention Networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6367–6371, Singapore, Singapore, 2022.

[8] Salvi, D., Bestagini, P., and Tubaro, S., "Synthetic Speech Detection through Audio Folding," in *ACM International Workshop on Multimedia AI against Disinformation (MAD)*, p. 3–9, 2023.

[9] Bishop, C. M. and Nasrabadi, N. M., *Pattern recognition and machine learning*, volume 4, Springer, 2006.

[10] Schmidt, M., Le Roux, N., and Bach, F., "Minimizing finite sums with the stochastic average gradient," *Mathematical Programming*, 162(1–2), pp. 83–112, 2017.

[11] Silverman, B. W., *Density estimation for statistics and data analysis*, Routledge, 2018.

[12] Wang, X. et al., "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, 64, 2020.

[13] European Broadcasting Union, *ECU R128-2023: Loudness Normalization and Permitted Maximum Level of Audio Signals*, tech.ebu.ch, 2023 [2010].

[14] Jia, F., Majumdar, S., and Ginsburg, B., "MarbleNet: Deep 1D Time-Channel Separable Convolutional Neural Network for Voice Activity Detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6818–6822, 2021.

[15] Müller, N. M., ieckmann, F., Czempin, P., Canals, R., Böttinger, K., and Williams, J., "Speech is Silver, Silence is Golden: What do ASVspoof-trained Models Really Learn?" in *Automatic Speaker Verification and Spoofing Countermeasures Challenge (ASVspoof)*, pp. 55–60, 2021.

[16] Cuccovillo, L., Gerhardt, M., and Aichroth, P., "Audio Transformer for Synthetic Speech Detection via Formant Magnitude and Phase Analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4805–4809, Seoul, Korea, 2024.

[17] Brümmer, N. and du Preez, J., "Application-independent evaluation of speaker detection," *Computer Speech and Language*, 20(2–3), pp. 230–275, 2006.

[18] Leeuwen, D. A. and Brümmer, N., *An Introduction to Application-Independent Evaluation of Speaker Recognition Systems*, p. 330–353, Springer, 2007.

[19] Kim, B. and Rafii, Z., "Lossy Audio Compression Identification," in *European Signal Processing Conference (EUSIPCO)*, pp. 2459–2463, Rome, Italy, 2018.

[20] *ENFSI Guideline for Evaluative Reporting in Forensic Science*, European Network of Forensic Science Institutes, 2015, version 3.0.