Detecting AI-Generated Speech: Synthetic Voice Classification using Deep Learning

Giuseppe Doda
University of Zurich
giuseppe.doda@uzh.ch

Ronald Domi
University of Zurich
ronald.domi@uzh.ch

Arjun Roy
University of Zurich
arjun.roy@uzh.ch

***Abstract*—The rapid advancement of text-to-speech (TTS) and voice conversion technologies has increased the realism of synthetic speech, raising security risks related to fraud, identity theft, and misinformation. This project addresses the challenge of distinguishing between bona fide (human) and spoofed (AI-generated) audio by implementing, evaluating and comparing three distinct deep learning architectures. We approach this classification problem from two perspective, image based models and time series models. The three models developed are: Convolutional Neural Network (CNN), Vision Transformer (ViT) and Long Short-Term Memory (LSTM) network. We utilize the ASVspoof 2021 dataset with labeled bona-fide or spoof audios to train and assess these models. Finally, we compare the results given by the three models used in the project.**

I. Introduction

The rapid development of synthetic speech generation has evolved greatly in recent years. While these technologies offer significant benefits for accessibility and entertainment, their ability to clone voices from as little as a few seconds of reference audio introduces its own set of problems. These tools have also the potential to be used for malicious actors to commit fraud, spread misinformation, and impersonate individuals through voice spoofing attacks.

The ASVspoof challenge series [1] has been very useful by providing standardized datasets and evaluation protocols where different models can be trained and evaluated in spoof-detection.

We formulate the core problem as a binary classification task: given an input audio sample, the system must determine whether it was produced by a human voice or synthesized by an algorithm.

II. Related Work

Voice spoofing detection has been extensively studied. Traditional approaches relied on hand-crafted features such as linear frequency cepstral coefficients (LFCCs) combined with classical machine learning classifiers. However, deep learning approaches have demonstrated superior performance in recent ASVspoof challenges.

Convolutional Neural Networks have been successfully applied to spectrogram-based analysis, treating audio classification as a computer vision problem. Vision Transformers, which have revolutionized image classification, have recently been adapted for audio tasks with promising results. Recurrent neural networks, particularly LSTMs, working on the raw audio time-series have also proved to be very competitive to the image based networks.

III. Methodology

A. Dataset

We utilize the ASVspoof 2021 Logical Access (LA) dataset [1], which contains audio samples of both bonafide human speech and synthetic speech. The datasets main concern is the imbalanced class representation, with approximately 90% spoof samples and only 10% bonafide samples.

To combat this particular problem, the pre-processing pipeline will use stratified split, so that all the training, validation and test sets have the same class distribution. In addition, we implement class weights, so we weigh bonafide 9x higher compared to spoof to compensate for class imbalance.

All audio files are sampled at 16 kHz. After studying the duration of all the samples on the dataset, we decided to crop or pad to 4 seconds. This makes the rest of the pipeline easier to work with, as handling variable input size takes more effort. 4 seconds, or 200 frames captures the 95th percentile or all the audio sample lengths.

The data after padding, cropping is normalized and augmented before being used by the models.

B. Feature Extraction

We employ two distinct feature representations based on whether we will use an image based model or the time-series representation.

1) Mel-Spectrograms: For the CNN and Vision Transformer models, we convert raw audio waveforms into log-scale Mel-spectrograms. The spectrograms are computed with the following parameters:

- Number of Mel bands: 128
- FFT size: 2048
- Hop length: 512 samples
- Frequency range: 20 Hz to 8000 Hz

The resulting spectrograms are treated as single-channel grayscale images with dimensions $128 \times T$, where T is the number of time frames (here 200). Each spectrogram is min-max normalized to the range $[0, 1]$ to facilitate neural network training.

2) MFCCs: For the Bi-LSTM model, we extract Mel-frequency cepstral coefficients (MFCCs). We compute 13 static MFCC coefficients. For the sake of simplicity, we did not include first-order (delta) and second-order (delta-delta) derivatives, resulting in 39 features per time frame, in our implementation.

The MFCC features are normalized per coefficient across the time dimension to have zero mean and unit variance.

C. Data Augmentation

To improve model generalization and prevent overfitting, we apply different augmentation strategies tailored to each feature representation:

1) Spectrogram Augmentation: For the CNN and ViT models operating on Mel-spectrograms, we apply SpecAugment:

- **Time masking**: Random masking of up to 30 consecutive time frames
- **Frequency masking**: Random masking of up to 20 consecutive frequency bins
- **Time Shifting**: Random move the audio in the time dimension, looping back the audio when passing the 200 frames limit.

Other augmentation techniques like adding noise, time stretching an volume gain were note implemented for sake of simplicity. Since we used fixed padding as the input, we skipped crop augmentation. It is important to note that this augmentation is done before normalization.

2) Audio Augmentation: For the Bi-LSTM model processing raw audio features, we apply:

- **Gaussian noise injection**: Addition of random Gaussian noise with amplitude factor 0.005
- **Volume scaling**: Scaling the volume (0.8 to 1.2)
- **Time shift**: The maximum shift the audio can have is 10% of the length

This augmentation simulates realistic recording conditions and microphone artifacts.

D. Model Architectures

1) CNN with Attention: Our CNN architecture consists of four convolutional blocks, each containing:

- 2D convolution (kernel size 3×3, stride 1, padding 1)
- Batch normalization
- ReLU activation
- Max pooling (2×2)

The number of channels progresses as: $1 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512$.

Global average pooling reduces spatial dimensions, and a dropout layer (rate 0.3) provides regularization. Finally, a linear classifier produces class logits.

2) Vision Transformer: Our Vision Transformer (ViT) architecture adapts the transformer paradigm to spectrogram classification. The key components are:

- **Patch embedding**: The input spectrogram is divided into non-overlapping 16×16 patches, which are linearly embedded into 768-dimensional vectors
- **Position encoding**: Learnable position embeddings are added to preserve spatial structure
- **Class token**: A learnable CLS token is prepended to the sequence
- **Transformer encoder**: 6 transformer layers with 8 attention heads each
- **Classification head**: A linear layer maps the final CLS token representation to class logits

Each transformer layer consists of multi-head self-attention followed by a feed-forward network (MLP ratio 4.0), with layer normalization and residual connections. Dropout (rate 0.1) is applied for regularization.

The self-attention mechanism allows the model to capture global dependencies across the entire spectrogram, potentially identifying subtle patterns that span multiple frequency bands and time frames.

3) Bidirectional LSTM: Our recurrent architecture processes the sequential MFCC features through:

- **Input projection**: Linear layer mapping 13 MFCC features to 256 dimensions, followed by layer normalization, ReLU activation, and dropout
- **Bidirectional LSTM**: 2 layers with hidden size 256, processing sequences in both forward and backward directions (effective hidden size 512)
- **Attention mechanism**: Temporal attention aggregates frame-level representations into a fixed-size context vector
- **Classification head**: Two-layer MLP with dropout, mapping the context vector to class logits

E. Handling Class Imbalance

We address the class imbalance through weighted cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} w_{y_i} \log(\hat{y}_i) \tag{1}$$

where w_{y_i} is the weight for the true class of sample i. We set $w_{\text{bonafide}} = 5.0$ and $w_{\text{spoof}} = 0.56$, forcing the model to pay substantially more attention to bonafide samples, while also preventing it from completely disregarding the majority spoof class.

F. Training Configuration

All models are trained with consistent hyperparameters to ensure fair comparison:

- **Optimizer**: Adam with learning rate 0.001 and weight decay 10^{-4}
- **Batch size**: 32
- **Maximum epochs**: 50
- **Learning rate scheduler**: ReduceLROnPlateau with patience 5 and factor 0.5
- **Early stopping**: Patience of 10 epochs based on validation loss

TABLE I: Performance comparison of the three architectures on the test set. Results show macro-averaged metrics.

Model	Acc.	Prec.	Rec.	F1
CNN + Attn	0.9891	0.9892	0.9891	0.9891
ViT	0.8974	0.8053	0.8974	0.8488
Bi-LSTM	0.9657	0.9666	0.9657	0.9661

TABLE II: Per-class performance metrics. Due to class imbalance, per-class analysis is crucial for understanding model behavior.

Model	Class	Prec.	Rec.	F1
CNN	Bonafide	0.9429	0.9516	0.9472
	Spoof	0.9945	0.9934	0.9939
ViT	Bonafide	0.0000	0.0000	0.0000
	Spoof	0.8974	1.0000	0.9459
Bi-LSTM	Bonafide	0.8162	0.8594	0.8373
	Spoof	0.8974	0.9779	0.9808

IV. Experimental Setup

A. Evaluation Metrics

Given the class imbalance, we report multiple metrics beyond simple accuracy:

- **Accuracy**: Overall classification accuracy
- **Precision**: Proportion of true positives among predicted positives
- **Recall**: Proportion of true positives among actual positives
- **F1-Score**: Harmonic mean of precision and recall

We compute both macro-averaged metrics (treating both classes equally) and per-class metrics to assess performance on both bonafide and spoof samples.

V. Results

A. Quantitative Performance

Table I presents the overall performance of all three models on the test set. All metrics are computed after training convergence with early stopping.

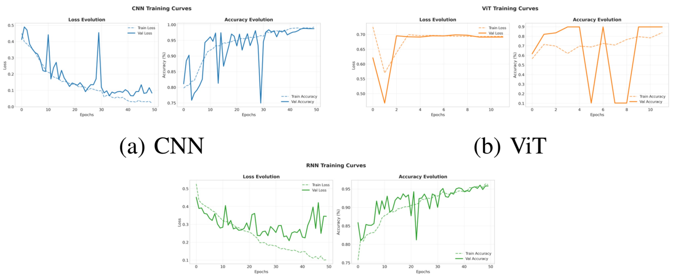
Note: Performance metrics will be populated after training completion. Placeholder dashes indicate pending experimental results.

B. Per-Class Performance

Table II breaks down the performance by class, which is particularly important given the class imbalance.

C. Training Dynamics

Figure 1 shows the training and validation loss curves. The convergence behavior varies across architectures, with the CNN typically converging faster due to its smaller parameter count, while the ViT requires more epochs to fully leverage its representational capacity.



(a) CNN

(b) ViT

(c) Bi-LSTM

Fig. 1: Training and validation loss curves for all three models. Early stopping prevents overfitting by monitoring validation performance.

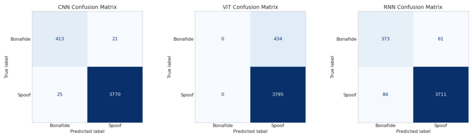


Fig. 2: Confusion matrix showing the classification performance across all models. Rows represent true labels, columns represent predictions.

D. Confusion Matrix

Figure 2 presents the confusion matrix for the models. This matrix reveals whether models exhibit bias toward the majority class (spoof) and their ability to correctly identify the minority class (bonafide).

E. Performance Metrics Visualization

Figure 3 provides a visual comparison of the key performance metrics across all three architectures, highlighting the relative strengths of each model.

F. Recall Analysis

Figure 4 shows the per-class recall for each model, which is particularly important given the class imbalance in the dataset.

G. ROC Curve Analysis

Figure 5 displays the ROC curves for all three architectures. The area under the curve (AUC) provides a single-number summary of model performance across all classification thresholds.

VI. Discussion

A. Architectural Insights

The comparison of three distinct architectures provides insights into the nature of synthetic speech artifacts:

a) CNN Performance: The convolutional approach treats spoofing detection as a visual pattern recognition problem. The attention mechanism enhances performance by allowing the model to focus on localized artifacts in the spectrogram. This architecture is particularly effective when synthetic speech contains visible spectral discontinuities or unnatural patterns.

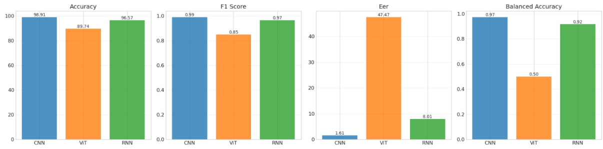


Fig. 3: Comparison of performance metrics (Accuracy, Precision, Recall, F1-Score) across all three architectures.

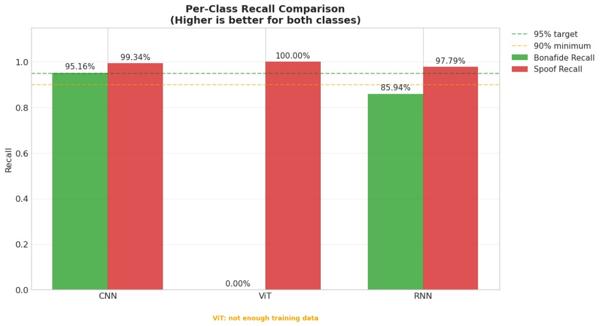


Fig. 4: Per-class recall comparison across the three models, showing performance on bonafide and spoof samples separately.

b) ViT Performance: The Vision Transformer's self-attention mechanism enables modeling of long-range dependencies across the spectrogram. This global perspective can capture subtle correlations between distant frequency bands and time frames that might be missed by local convolutions. However, this comes at the cost of increased computational requirements and potentially slower convergence.

c) Bi-LSTM Performance: The recurrent architecture operates on MFCC features, modeling the temporal dynamics of speech production. The bidirectional processing captures both past and future context, while the attention mechanism identifies critical time frames. This approach is effective at detecting temporal anomalies in synthetic speech, such as unnatural prosody or irregular timing patterns.

B. Class Imbalance Impact

The severe class imbalance (90/10 split) requires careful consideration. Our weighted loss approach prevents the model from simply predicting the majority spoof class most of the time. However, this introduces a hyperparameter (the weight ratio) that must be tuned. We believe our approach is robust while acknowledging alternatives such as oversampling, undersampling, and focal loss could also be explored.

C. Generalization Considerations

A key challenge in spoofing detection is generalization to unseen synthesis methods. The ASVspoof 2021 dataset contains multiple TTS and voice conversion systems, but new synthesis techniques are constantly emerging. Our data augmentation strategies aim to improve robustness, but evaluation on completely novel synthesis methods remains an important future direction.

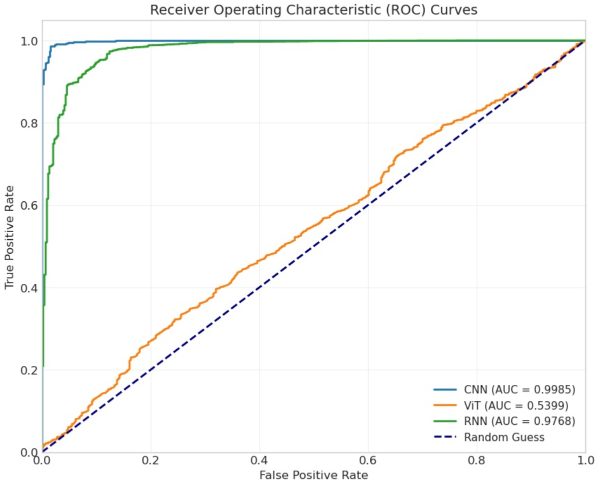


Fig. 5: Receiver Operating Characteristic (ROC) curves for all three models, illustrating the trade-off between true positive rate and false positive rate.

VII. Conclusion

This paper presented a comprehensive comparison of three deep learning architectures for detecting AI-generated speech. We implemented a CNN with attention, a Vision Transformer, and a Bidirectional LSTM, each operating on different feature representations (Mel-spectrograms and MFCCs). Through careful handling of class imbalance and appropriate data augmentation, we trained these models on the ASVspoof 2021 dataset.

Our analysis reveals the strengths and limitations of each approach:

- CNNs excel at detecting local spectral artifacts through convolutional filters
- Vision Transformers capture global dependencies via self-attention mechanisms
- Bi-LSTMs model temporal dynamics and sequential patterns in acoustic features

The comparative evaluation provides insights into which architectural choices are most effective for this critical security task. As synthetic speech becomes increasingly sophisticated, robust detection methods are essential for protecting voice-based authentication systems, preventing fraud, and maintaining trust in audio content.

The code and models developed in this work are available to support reproducibility and future research in this important area.

References

[1] H. Delgado, N. Evans, T. Kinnunen, K. A. Lee, X. Liu, A. Nautsch, J. Patino, M. Sahidullah, M. Todisco, X. Wang, and J. Yamagishi, "ASVspoof 2021 Challenge - Logical Access Database (1.1)," 2021.