# AUDIO DEEPFAKE DETECTION WITH SONIC SLEUTH MODEL

**[1]ANFAL SULTAN ALSHEHRI, [2]DANAH SALEH ALMALKI SUPERVISOR, [3]SOMAYAH ALBARADEI**

[1,2,3] Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
E-mail: Alshehrianfal,Danaalmalkexvii@gmail.com

**Abstract -** Information dissemination and preservation are crucial for societal progress, especially in the technological age. While technology fosters knowledge sharing, it also risks spreading misinformation. Audio deep fakes, convincingly fabricated audio using Artificial intelligence (AI), exacerbate this issue. We present Sonic Sleuth, an AI model for detecting audio deepfakes. Our approach leverages Deep Learning (DL) for a robust detection model. Meticulous data preprocessing and rigorous experimentation with various models led to the implementation of the most effective solution with a custom CNN model. Our comprehensive testing resulted in a highly accurate model (98.27% accuracy, 0.016 EER) trained on a substantial dataset of real and synthetic audio. In addition to an 84.92% accuracy and 0.085 EER on an external dataset, these results demonstrate Sonic Sleuth's potential as a powerful tool against audio misinformation.

**Keywords -** Artificial Intelligence – Deepfake Detection – Machine Learning – Deep Learning – Deepfake Audio – Audio Forensics

## I. INTRODUCTION

The cybersecurity landscape is constantly evolving, and a novel and highly deceptive threat has emerged: deepfakes.Unlike traditional cyberattacks that often necessitate significant technical prowess, deepfakes leverage the power of artificial intelligence (AI) to create hyperrealistic and convincing fabricated audio, video, or text content. This intrinsic characteristic makes them particularly perilous, as they can effortlessly deceive unsuspecting victims, potentially leading to devastating consequences.

The threat posed by deepfakes is significant and multifaceted, impacting individuals, organizations, and societies. Developing robust deepfake detection methods is crucial for mitigating these risks. By harnessing the power of deep learning,we can create effective tools to identify and combat deepfakes, safeguarding the integrity of digital content and maintaining public trust. This research addresses a critical challenge in the digital age and contributes to the broader efforts of enhancing cybersecurity and information authenticity.

## II. BACKGROUND

### 2.1 Deepfake Generation

Generating DFs is accomplished through a generative adversarial network (GAN), which consists of two parts: Convolutional Neural Networks (CNN) and Deconvolutional Neural Networks (DNN), as shown in Figure 1. They aim to produce data that looks just like real data. The CNN is the generator model, while the DNN is the discriminator model. The generator is trained to produce data, and the discriminator is trained to distinguish between data generated by the generator and real data. When the discriminator successfully distinguishes between real and generated data, the generator generates an enhanced version of the data until the discriminator cannot differentiate between the generated and real data.

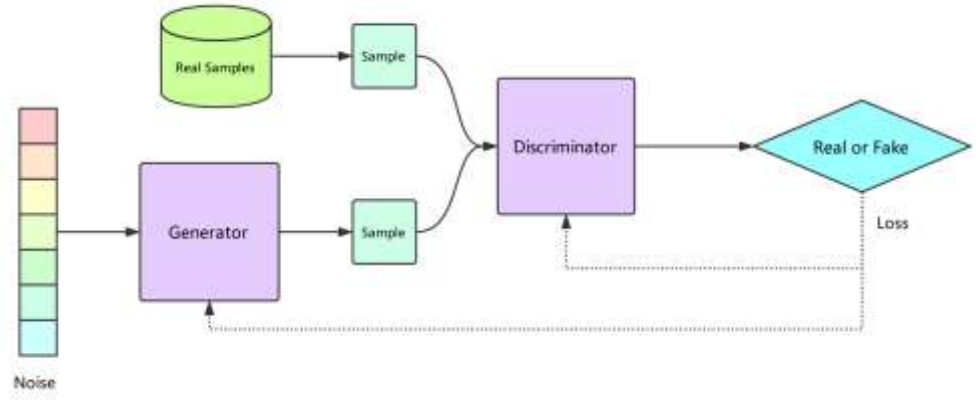



**Figure 1. Generative AI structure**
**Gu et al. (2018)**

### 2.2 Time Domain and Frequency Domain

The time domain represents a sound wave using a function in time where a graph of such a domain shows the change in the signal value (amplitude) over time Camastra & Vinciarelli (2015). The time domain gives us information about the wave frequency, wavelength, and amplitude. Various calculations can be made to analyze and understand a signal, including Time domain filtering, feature extraction, and signal visualization Camastra&Vinciarelli (2015). Another representation of a signal is the frequency domain, where instead of showing amplitude change over time, it shows how much of a signal lies in a specific frequency over a range of frequencies Tenoudji (2018). Two types of frequency domain graphs are the spectrum and the spectrogram. A spectrum captures the signal's amplitude vs. frequency distribution at a specific point in time, while a spectrogram shows the same information across a period of time; it plots the frequency of a wave over time and uses colors to indicate the amplitude of each frequency, the brighter the color, the higher the amplitude Natsiou & O'Leary (2022).

The Fourier Transform, a powerful mathematical tool, is used to convert signals from the time domain to the frequency domain Tenoudji (2018). This transformation breaks down a signal into the different waves of a constant frequency that make up the whole signal. It provides us with the spectrum of an acoustic wave Tenoudji (2018). The spectrum can then be used to obtain a spectrogram by combining the spectra of multiple overlapping audio segments into a single plot of frequency against time Natsiou & O'Leary (2022).

### 2.3 Literature Review

Research in the field of DF audio detection is a continuous effort to defend against the evolving techniques of generative AI. And just like any AI task, robustness is a main goal of DF research, which requires AI to move to the next level of deep understanding Marcus (2020).

One way to achieve that depends on the training data of AI models. In the case of DF audio detection, a common approach to achieve better generalization of detection is to combine multiple datasets, not only to train your model on both fake and real audio but also to train on data of different nature, such as a dataset that was meant for spoofing attacks or in-the-wild datasets that evaluate your model on realistic audios Frank & Schönherr (2021). The dataset WaveFake Frank & Schönherr (2021) is one example where 6 different architectures and a text-to-speech pipeline were used to generate fake audio, addressing the problem of detection models generalization to different DF audio generation methods.

A similar approach was that of Kawa et al. (2022), where the "AttackAgnostic Dataset"was proposed to evaluate the generalization of models. The dataset combined two deepfake datasets and one audio spoofing dataset: WaveFake, FakeAVCeleb, and ASVspoof2019(LA subset), respectively. Spoofing attacks have different elements than DF audio since the aim is to deceive Automatic Speech Verification (ASV) systems instead of humans Kawa et al. (2022). Spoofing audio also contains distortion and background noises that aim to confuse ASV systems. These subtle differences can be useful for increasing a model's accuracy and ability to generalize to new real-life data Kawa et al. (2022).

Additionally, Kawa et al. (2022) investigated the effect of different front-ends (features) on DF audio detection results. This came from the idea that DF audio generation models pay attention to the details and features of the audio that are within the hearing scope of humans; therefore, features that are not audible to humans, i.e., of the higher frequency range, might be a giveaway that distinguishes synthesized audio from genuine Kawa et al. (2022). Architectures trained on Linear frequency cepstral coefficients (LFCC) front–end performed better than those trained on features that are within the human hearing scope, such as mel–frequency cepstral coefficients (MFCC) and spectrogram–based features Kawa et al. (2022).

Another approach feature-wise is taking advantage of vocoders as in Sun et al. (2023), a neural network that takes the features of an acoustic wave, e.g., Mel-spectrogram, as an input and outputs a waveform Zhang et al. (2023). Since vocoders are commonly the last

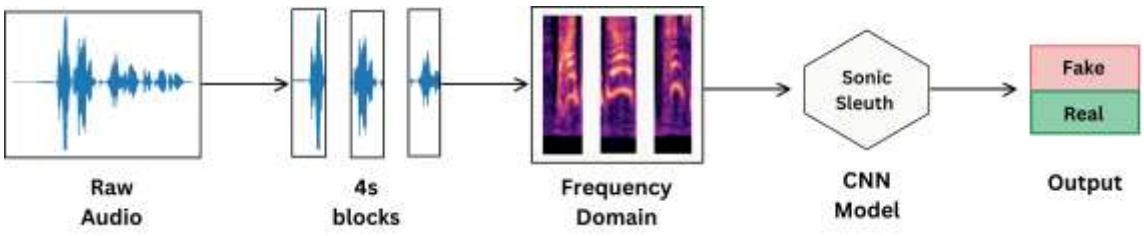



**Figure 2. Deepfake Detection Approach**

stage in Speech Synthesis frameworks, identifying traces of vocoder's artifacts can help detect DF audios Sun et al. (2023). This is possible by training a model to identify the vocoder's artifacts in audio

before going through a second model that detects audio DF Sun et al. (2023).

## III. APPROACH

This section discusses our approach to developing the audio DF detection models, from obtaining the data to preprocessing, feature extraction methods, and training details. Figure 2 shows the generalapproach we will use to detect deepfakes.

### 3.1 Datasets

The dataset is a crucial element in deep learning, necessitating a thoughtful and informed selection process that considers the specific task and model requirements. For deepfake audio detection, two main types of data are required: generated audio (fake) and human voice audio (real).We trained ourCNNmodel on three datasets:ASVspoof, In theWild, and FakeAVCeleb, which combine both types and a total of 70,000 samples. And to ensure the generalization of our model,we further tested it on the Fake-Or-Real Dataset.

- ASVspoof2019 ASVspoof (Automatic Speaker Verification Spoofing and Countermeasures) is an international challenge focusing on spoofing detection in automatic speaker verification systems Wang et al. (2020). The ASVspoof2019 dataset contains three subdatasets: Logical access, Physical access, and Speech deepfake. Each of these datasets was created using different techniques depending on the task, like text-to-speech (TTS) and voice conversion (VC) algorithms Wang et al. (2020). We will be utilizing the train subset of the logical access dataset in our experiment.
- In the Wild is a dataset containing fake and real audio from 58 celebrities and politicians sourced from publicly available media. It was designed to evaluate deepfake detection models, particularlytheir ability to generalize to real-world data. Müller et al. (2022)
- FakeAVCeleb contains both deepfake and real audio and videos of celebrities. The fake audio was created using a text-to-speech service followed by manipulation with a voice cloning tool to mimic the celebrity's voice Khalid et al. (2021). We extracted the audio as WAV files from each video.
- The Fake-or-Real (FoR) Dataset comprises 111,000 files of real speech and 87,000 files of fake speech. It encompasses both MP3 and WAV file formats. offers four distinct versions to suit various needs. The 'for-original' files are in their original state, while the 'for-norm' version with normalization. 'For-2sec' is shortened to 2 seconds, while 'for-rerec' simulates re-recorded data, depicting deepfake from a phone call scenario Abdeldayem (2022).

Combining the first three diverse datasets for training and reserving the last one for testing, we aim to achieve a comprehensive and varied dataset for our task. Further details are illustrated in Table 1.

| Name | Size | | Length | Sample Rate | File Format | URL |
|---|---|---|---|---|---|---|
| | real | fake | | | | |
| ASVspoof2019 | 2,580 files | 22,800 files | Avg. 3 s | - | flac | Link |
| 'In-the-Wild' | 19,963 files | 11,816 files | Avg. 4.3 s | 16 kHz | WAV | Link |
| FakeAVCeleb | 10,209 files | 11,357 files | Avg. 5 s | - | MP4 | Link |
| The Fake-or-Real (FoR) Dataset | 111,000 files | 87,000 files | (1 - 20)s | - | WAV / MP3 | Link |

**Table 1. Datasets for model development**

### 3.2 Data Prepossessing

Several essential preprocessing steps are applied to our datasets to ensure uniformity and compatibility of the audio data with our machinelearning models. including:

- First, any silence in the audio was trimmed to enable the model to focus on speech parts of the audio.
- Second, all audio clips are trimmed to a standardized length of 4 seconds or padded by repeating the audio.
- Third, the audio data is downsampled to a rate of 16 kHz. Downsampling reduces the data size and computational load without significantly compromising the audio signal's quality or essential characteristics.
- Lastly, the audio is converted to a mono channel using channel averaging. Stereo audio, which uses two channels (left and right), is averaged into a single channel, resulting in a mono audio signal.

These conversions simplify the data and ensure a uniform input format for the machine-learning models, which is particularly important when dealing with diverse audio sources.

### 3.3 feature extraction

Instead of working with raw audio files, a common AI practice is converting audio into a visual representation. Subsequently, image classification models like CNN can be utilized for audio classification. In this study, we chose to convert the audio to spectrograms in the frequency domain using

different feature extraction techniques, including Linear Frequency Cepstral Coefficients (LFCC), Mel-Frequency Cepstral Coefficients (MFCC), and Constant-Q Transform (CQT).

LFCC and MFCC are derived from a signal's short-time power spectrum and provide compact representations of spectral characteristics, capturing essential features for further analysis. CQT, on the other hand, is particularly effective in analyzing audio signals with a wide range of frequencies, offering a logarithmically spaced frequency analysis that mimics human auditory perception. These feature extraction methods enhance our ability to detect subtle patterns and anomalies in audio data, improving the accuracy of our deepfake detection models.

we plan to compare the effects of these different features on the model's ability to learn features and generalize the results to new, unseen data.

### 3.4 Sonic Sleuth structure and Training details

We employ a Convolutional Neural Network (CNN) to classify processed audio samples into real or synthesized audio. The structure includes 3 convolutional layers, each followed by a max-pooling layer; after the convolutional layers, the output is flattened and passed through two fully connected (dense) layers and a 10% dropout rate to prevent overfitting. The final dense layer has a single output unit with a sigmoid activation function, providing a probability score for classifying the input as deepfake. Figure 3 illustrates the architecture of our custom CNN model.

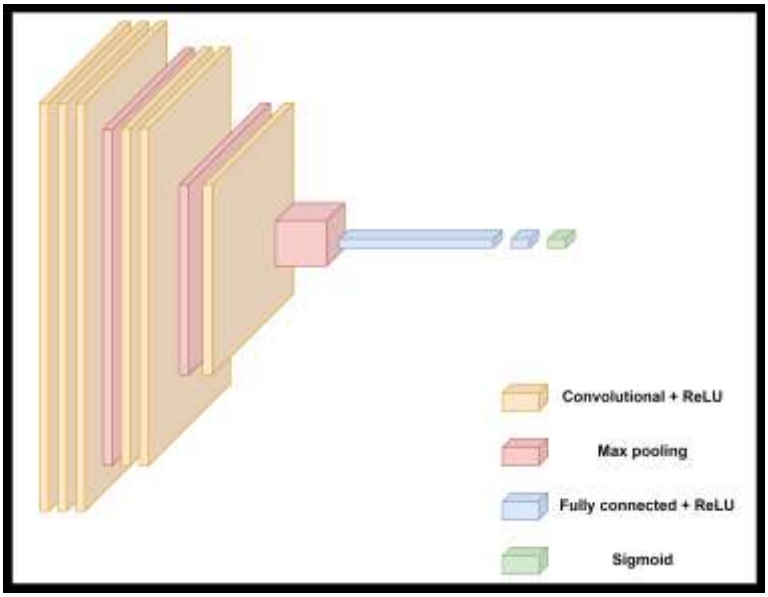



**Figure 3. Sonic Sleuth Architecture**

For training the deepfake audio detection models, the dataset was split into train, validation, and test subsets with a ratio of 8.5:1:0.5,respectively.

The models were trained for 100 epochs using the following configurations:

1. Early Stopping: An early stopping technique was implemented with a patience of 10 epochs to prevent overfitting. This means the training process would be stopped if the validation loss did not improve for 10 consecutive epochs.
2. Optimizer: The ADAM optimizer was used for updating the model's weights during training.
3. Loss Function: Binary cross-entropy loss was used as the loss function, which is suitable for binary classification problems.
4. Class Weights: Class weights were calculated and applied during the training process to handle the class imbalance in the dataset. Class weights adjust the importance of each class during the loss calculation, helping the model learn from imbalanced datasets more effectively while avoiding bias.

By incorporating these techniques and configurations, the model was trained to learn discriminative features from the audio spectrograms and classify them as real or deepfake with improved performance and generalization to new, unseen data.

| Feature | EER | Accuracy | F1 |
|---|---|---|---|
| CQT | 0.0757 | 94.15% | 95.76% |
| LFCC | 0.0160 | 98.27% | 98.65% |
| MFCC | 0.0185 | 98.04% | 98.45% |

**Table 2. Model Performance Metrics**

| Feature | EER | Accuracy | F1 |
|---|---|---|---|
| CQT | 0.0942 | 82.51% | 83.19% |
| LFCC | 0.1718 | 75.28% | 72.63% |
| MFCC | 0.3165 | 61.24% | 54.94% |

**Table 3. Model Performance Metrics on External Dataset**

## IV. RESULTS AND DISCUSSION

To accurately represent the results of our experiments, we used several evaluation metrics, including balanced accuracy, F1 score, and Equal Error Rate (EER).

Table 2 presents the results of testing our models on the test subset of our dataset. LFCC demonstrated the best performance across all metrics, achieving an EER of 0.0160, an Accuracy of 98.27%, and an F1 Score of 98.65%. MFCC also performed well, with an EER of 0.0185, an Accuracy of 98.04%, and an F1 Score of 98.45%. In contrast, CQT exhibited the lowest performance, with an EER of 0.0757, an Accuracy of 94.15%, and an F1 Score of 95.76%. These results indicate that LFCC and MFCC are superior feature extraction methods for this specific classification task, providing higher accuracy and lower error rates compared to CQT.

### 4.1 Test on External Dataset

To assess the generalizability of our models, we tested them on an external dataset, Fake-or-Real (FoR). The testing was conducted on a balanced subset of 2116 samples of 4-second audio. The performance metrics are reported in Table 3.

While the accuracy of the models has slightly decreased compared to the initial evaluation, it's noteworthy that CQT, which previously had the lowest accuracy among the three methods, now showcases improved performance. With an EER of 0.0942, an Accuracy of 82.51%, and an F1 Score of 83.19%, CQT has made significant strides, indicating its viability as a feature extraction method even in diverse datasets.

Similarly, LFCC, despite experiencing a decline in performance, continues to exhibit its utility in real-world scenarios. LFCC, with an EER of 0.1718, an Accuracy of 75.28%, and an F1 Score of 72.63%, maintains its effectiveness in capturing relevant audio features.

Meanwhile, MFCC, with an EER of 0.3165, an Accuracy of 61.24%, and an F1 Score of 54.94%, appears not to be the best for real-life applications due to weak generalizability.

### 4.2 Ensemble Approach

In machine learning, the ensemble model approach combines the output of multiple models that are trained for the same task, which can reduce error. In this experiment, we tested one type of ensemble model: obtaining the output of multiple models on the same sample and then combining the outputs to get the final results.

| Feature | EER | Accuracy | F1 |
|---|---|---|---|
| CQT and LFCC | 0.0851 | 84.92% | 84.73% |

**Table 4. Model Performance Metrics on External Dataset (Ensemble Approach)**

Table 4 showcases the results of the ensemble approach that combines CQT and LFCC models. The combination of CQT and LFCC on the external dataset presents promising results, with an Equal Error Rate (EER) of 0.0851, an Accuracy of 84.92%, and an F1 Score of 84.73%. This joint approach showcases slight improvement compared to the individual methods, indicating the complementary nature of CQT and LFCC in capturing relevant audio features. These results provide valuable insights for further exploration and optimization, highlighting the potential for synergy between different feature extraction methods to achieve superior performance across diverse datasets and application contexts.

## V. CONCLUSION

In conclusion, our research centered on the development of an AI model named Sonic Sleuth for the detection of audio deepfakes. Utilizing a custom convolutional neural network (CNN), we trained our model on three diverse datasets: ASVspoof, In the Wild, and FakeAVCeleb, amassing a total of 78,725 audio samples. To prepare the data, we employed feature extraction techniques like Linear Frequency Cepstral Coefficients (LFCC), Mel-Frequency Cepstral Coefficients (MFCC), and Constant-Q Transform (CQT) to convert audio signals into spectrograms for detailed analysis.

Our study demonstrated that LFCC performed optimally on the training dataset, achieving an EER of 0.0160 and an accuracy of 98.27%. Conversely, CQT showed superior performance on the external dataset, indicating better generalization with an EER of 0.0942 and an accuracy of 82.51%. Moreover, the ensemble approach combining CQT and LFCC further improved performance, yielding an EER of 0.0851 and an accuracy of 84.92%.

For future work, we propose further exploration of various ensemble methods to boost performance. Integrating a broader range of features could help

identify the most complementary combinations, thereby enhancing the model's performance and generalization capabilities even further.

## ACKNOWLEDGEMENTS

This work was written by Anfal Alshehri and Dana Almalki, under the supervision of Dr.Somayah Albaradei. We are grateful for the guidance and support provided throughout this project.

## REFERENCES

[1] Abdeldayem M., 2022, The Fake-or-Real Dataset, https: //www.kaggle.com/datasets/mohammedabdeldayem/ the-fake-or-real-dataset
[2] Camastra F., Vinciarelli A., 2015, Machine learning for audio, image and video analysis: Theory and applications. Springer
[3] Frank J., Schönherr L., 2021, CoRR, abs/2111.02813
[4] Gu Y., Chen Q., Liu K., Xie L., Kang C., 2018. , doi:10.1109/ISGT.2019.8791575
[5] Kawa P., Plata M., Syga P., 2022, in Interspeech 2022. ISCA, doi:10.21437/interspeech.2022-10078, https://doi.org/10.21437% 2Finterspeech.2022-10078
[6] Khalid H., Tariq S., Kim M., Woo S. S., 2021, in Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2). https://openreview.net/forum?id=TAXFsg6ZaOl
[7] Marcus G., 2020, CoRR, abs/2002.06177
[8] Müller N. M., Czempin P., Dieckmann F., Froghyar A., Böttinger K., 2022, Interspeech
[9] Ntsiou A., O'Leary S., 2022, Audio representations for deep learning in sound synthesis: A review (arXiv:2201.02490)
[10] Sun C., Jia S., Hou S., AlBadawy E., Lyu S., 2023, Exposing AI-Synthesized Human Voices Using Neural Vocoder Artifacts (arXiv:2302.09198)
[11] Tenoudji F. C., 2018, Analog and digital signal analysis from basics to applications. Springer International Publishing
[12] Wang X., et al., 2020, ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech (arXiv:1911.01601)
[13] Zhang C., Zhang C., Zheng S., Zhang M., Qamar M., Bae S.-H., Kweon I. S., 2023, A Survey on Audio Diffusion Models: Text To Speech Synthesis and Enhancement in Generative AI (arXiv:2303.13336)

★★★