DEEP FAKE AUDIO DETECTION USING ZERO TRUST SECURITY

Ariba Shafaqat
Department of Computer Science
University of Engineering and Technology Lahore
Pakistan
aribashafaqat4@email.com

Wajeeha Javaid
Department of Computer Science
University of Engineering and Technology Lahore
Pakistan
malikwajeeha489@email.com

***Abstract*—The ability to identify deepfake content has become crucial in the era of developing technology, especially in the audio domain. Deepfake audio, which manipulates real audio to create misleading information, poses serious concerns to security, media, and public trust. To address this challenge, we propose a system for detecting deepfake audio using a custom-built Convolutional Neural Network (CNN), named DeepSonarCNN. It is designed to examine audio patterns and successfully spot irregularities that are typical of deepfake recordings.In order to ensure that the model can accurately distinguish between real and fake audio samples, we trained it on a comprehensive dataset called the Fake-or-Real (FoR) Dataset. This dataset includes data from the most recent TTS solutions, such as Deep Voice 3 and Google Wavenet TTS, as well as a variety of real human speech, including the Arctic Dataset and others. With a 97% classification accuracy, DeepSonarCNN efficiently detects anomalies in audio signal Mel spectrogram representations. In addition to model performance, our system incorporates the principles of Zero Trust Security (ZTS), which include access control and encrypted model protection utilising Fernet and AES.**

I. Introduction

The rapid growth of the Artificial Intelligence field in the previous years has enabled the easy creation of highly realistic fake media, or deepfakes. While so much focus is placed on fake videos, deepfake audio is no less worrying and has emerged as an equally serious concern. With the capabilities of Text-to-Speech (TTS) and voice cloning technology, deepfake audio systems can generate speech that closely resembles the voice, tone, and rhythm of real people very convincingly.This poses serious threats in domains such as cybersecurity, digital forensics, social engineering, and the dissemination of inaccurate information. Advanced TTS models like Google WaveNet, Deep Voice 3, and Tacotron can generate human-like speech that is nearly indistinguishable from actual human recordings. However, the same advancement in technology makes it far more difficult to identify fabricated audio content, especially when a great deal of synthetic effort is used to make the sample emotionally and contextually authentic. Traditional methods, which use manually picked audio features or basic machine learning models, often fail to detect deepfake audio from different voice generators or new speakers.The gaps in existing solutions emphasize the necessity of adaptable detection frameworks that can reliably deal with new challenges posed by modern synthesis methods. To address this problem, we propose DeepSonarCNN, a convolutional neural network model specifically designed for deepfake audio detection. Our model processes spectrogram representations of audio input and learns hierarchical features that highlight the minute differences between authentic and synthetic speech. Unlike earlier methods that depend on certain audio signals, DeepSonarCNN autonomously extracts patterns directly from the data, offering improved generalization and scalability.

In addition to detection capabilities, our system integrates Zero Trust Security principles to enforce strict identity verification and prevent unauthorized access to the model pipeline. We secured the model using fernet encryption. The model was trained on a well-curated dataset made up of real and fake speech. Real samples were collected from publicly available datasets such as CMU Arctic, LJSpeech, and VoxForge, with some other samples being recorded manually and fake samples were generated with high-quality TTS systems to make the training set diverse and challenging. The results of our evaluation demonstrate that DeepSonarCNN achieves a classification accuracy of 97%, showing superior performance in identifying deepfake audio compared to conventional methods. This paper contributes to the field of audio forensics by presenting a reliable deep learning approach that can handle the fast-changing techniques used in synthetic speech generation.

II. Related Work

A number of recent studies have investigated machine learning and deep learning techniques for identifying synthetic audio, especially using benchmark datasets such as Fake-or-Real (FoR).

The Sonic Sleuth model, a CNN-based system presented by Mahmud et al.[1] obtained a 0.016 equal error rate and 98.27% accuracy on the FoR dataset. Despite the model's remarkable detection capabilities, security features like access control and encryption were not integrated.

Wang et al. (2024) proposed MFAAN, a Multi-Feature Audio Authenticity Network combining MFCC, LFCC, and Chroma-STFT representations. Their parallel feature approach improved generalizability, though it introduced architectural complexity and did not address model security or data integrity.[2].

TABLE I
COMPARISON OF RELATED WORK IN DEEPFAKE AUDIO DETECTION

System / Model	Key Features	Gaps / Limitations	Security Considerations	Author(s) & Year
MFCC + ML Classifiers	Uses MFCC features with SVM, Random Forest, etc.	Basic models; struggles with generalization on complex audio fakes	No encryption or system-level security	Shaik et al., 2023
MFAAN	Multi-feature input (MFCC, LFCC, Chroma-STFT); parallel deep learning paths	Complex design; not lightweight for deployment	No mention of encryption or Zero Trust	Wang et al., 2024
Sonic Sleuth	Custom CNN; 98.27% accuracy on FoR dataset; low EER	No user access control or model protection	No Zero Trust model; lacks encryption	Mahmud et al., 2024
Survey by Yi et al.	Overview of datasets, features, models in deepfake audio detection	Literature review only; no implementation proposed	Highlights need for secure detection, but none proposed	Yi et al., 2023
Proposed Work (DeepSonar + ZT)	CNN for spectrograms; AES & Fernet encryption; login + OTP + MySQL logging	Requires GPU for real-time use; relies on structured input	Full Zero Trust architecture; AES + Fernet security integration	**This Work, 2025**

Shaik et al. (2023) developed a machine learning-based deepfake detector using MFCC features with classifiers like SVM and Random Forest. Despite its simplicity, the model achieved reasonable performance but lacked scalability and security considerations [3].

A separate line of research by Yi et al. (2023) provided a comprehensive survey of datasets, features, and architectures used in audio deepfake detection. The review emphasized the need for robust, secure, and generalizable detection systems to handle diverse synthesis techniques [4].

While these approaches focus primarily on model performance, few incorporate system-level safeguards. In contrast, our proposed system introduces a dedicated CNN architecture (DeepSonarCNN) optimized for spectrogram-based classification and integrates Zero Trust Security principles. This includes AES encryption for audio data, Fernet encryption for model protection, OTP-based access control, and secure-by-design practices, ensuring both detection accuracy and robust deployment security.

III. DATASET OVERVIEW

To evaluate the performance of our system, we used the Fake-or-Real (FoR) dataset, a large-scale corpus containing more than 195,000 audio utterances including both real and synthetic speech. The dataset was introduced by Reimao and Tzerpos.[5], and has been widely used for benchmarking deepfake audio detection systems.This dataset is especially designed for training and benchmarking speech deepfake detectors.

A. Data Sources

The FoR dataset gathers audio samples from both real human speech and state-of-the-art synthetic speech generators. The real speech data is gathered from well-established public datasets, such as the CMU Arctic[5], LJSpeech[6], and VoxForge[7] as well as some extra recordings contributed by the dataset authors. On the synthetic side, the dataset includes samples generated by using cutting-edge TTS technologies like Deep Voice 3 and Google WaveNet.

1) for-original Version: The `for-original` version includes raw audio samples directly collected from various sources without any normalization or preprocessing. This version is useful for evaluating model performance under unstandardized and diverse input conditions.

2) for-norm Version: The `for-norm` version is obtained from the original data and gets normalized in are in terms of sample rate, volume, and channels. It also maintains balance across gender and class compositions, offering a more controlled learning environment.

3) for-2sec Version: The `for-2sec` version is a truncated variant of the normalized dataset, where each audio clip is limited to a maximum duration of 2 seconds. This design simulates real-world scenarios where only short utterances are available, such as in security systems or short voice commands.

4) for-rerec Version: The `for-rerec` version builds upon the `for-2sec` subset but involves rerecording the audio through physical devices or communication channels. This version is intended to reflect realistic conditions, including distortions and noise introduced by transmission over phone lines or messaging platforms.

TABLE II
SUMMARY OF FOR DATASET VERSIONS

Dataset Version	Description
for-original	Raw audio from various sources; unprocessed.
for-norm	Normalized and class/gender balanced.
for-2sec	Truncated version with 2-second clips.
for-rerec	Rerecorded audio for real-world simulation.

IV. METHODOLOGY

To detect deepfake audio, we developed a custom Convolutional Neural Network called DeepSonarCNN, designed to operate on time-frequency representations of speech signals. The approach begins with transforming raw audio into Mel-spectrograms through a structured preprocessing pipeline. These spectrograms are then passed through a multi-layer CNN to classify the input as either real or synthetic.

A. Preprocessing

All audio files are first resampled to a fixed sampling rate of 16 kHz to ensure uniformity. Each clip is then either padded or truncated to a standard length of three seconds to maintain consistent input dimensions. The Short-Time Fourier Transform (STFT) is applied to convert the raw waveform into a time-frequency domain and then by a Mel-filter bank transformation using 64 filters. The resulting Mel-spectrogram is converted to the decibel (dB) scale and normalized to reduce variation across samples. These spectrograms, which highlight both spectral and temporal information are then used as input to the DeepSonarCNN model.

B. Model Architecture

The DeepSonarCNN model processes Mel-spectrograms through a series of convolutional blocks designed to extract time-frequency features that distinguish synthetic speech from real human speech. As shown in Figure 1, the architecture begins with a 2D convolutional layer with 16 filters and a kernel size of 3×3.

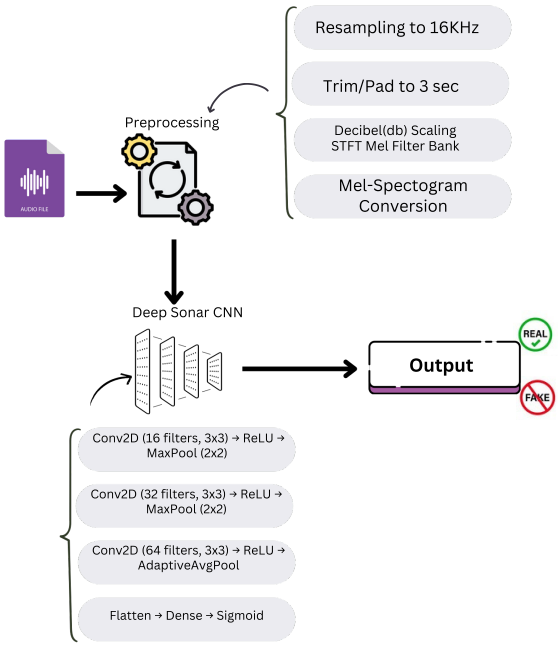

Fig. 1. DeepSonar CNN Achitecture

This layer captures low-level acoustic features such as sharp frequency edges or localized spectral patterns. A ReLU activation function follows to introduce non-linearity, enabling the network to model complex relationships. A 2×2 max pooling operation is applied to downsample the feature map and reduce computational load.

The second block increases the number of filters to 32 and repeats the convolution, ReLU, and max pooling structure. This layer captures high level features, including voice tone, prosody, and harmonics, usually altered a bit in deepfake audio.

In the third convolutional block, the model uses 64 filters with a 3×3 kernel, followed by ReLU activation. Instead of max pooling, this block uses an adaptive average pooling layer to compress each feature map into a single representative value, making the model more robust to variable-length inputs.

The pooled output is then flattened and passed into a fully connected dense layer, which integrates the learned features. A sigmoid activation function at the output layer provides a binary probability, with values closer to 0 indicating real audio and values near 1 indicating fake audio.

In this training process, we use the binary cross-entropy loss function, which is commonly used for tasks where the model needs to classify between two categories (like ”Fake” vs ”Real”). The optimizer used is Adam, which is a popular choice for training deep learning models, and we set the learning rate to 0.001. This learning rate controls how much the model’s weights are adjusted with each update.

The training runs for a total of 10 epochs. An epoch is one complete pass through the entire training dataset. In each epoch, the model is optimized, and by the end of the 10 epochs, the model has seen and learned from the data multiple times. The batch size is set to 16, meaning that in each step of training, 16 samples of data are processed at a time before the model updates its weights.

To help avoid the model from ”overfitting” (learning the training data too well and performing poorly on new data), we use a technique called early stopping. This method monitors the model’s performance on the validation set during training. As shown in Figure 2, the model’s accuracy on the validation set improves over time, indicating that the model is learning and generalizing well from the training data.

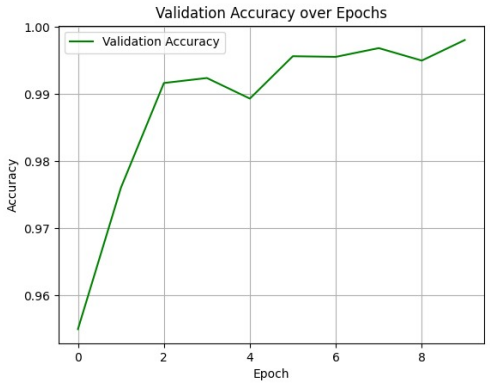

Fig. 2. Validation Accuracy over Epochs

C. Evaluation Metrics

The model was evaluated using standard classification metrics including precision, recall, F1-score, and accuracy. As shown in Table III, the DeepSonarCNN achieved an overall accuracy of 97% on the validation set. The model attained a precision of 1.00 for detecting fake audio, and a recall of 1.00 for identifying real audio, indicating a strong and balanced performance across both classes.

TABLE III
CLASSIFICATION REPORT OF DEEPSONARCNN

Class	Precision	Recall	F1-Score	Support
Fake	1.00	0.95	0.97	5368
Real	0.95	1.00	0.97	5406
Accuracy		0.97		10774
Macro Avg	0.98	0.97	0.97	10774
Weighted Avg	0.98	0.97	0.97	10774

V. MODEL EVALUATION AND OUTCOMES

To evaluate the trained model, we tested it on unseen audio samples. These inputs were first processed using the same preprocessing steps described earlier, including resampling, trimming, and Mel-spectrogram conversion. The processed spectrograms were then passed to the model for prediction. For each input, the model produced a binary classification result—labeling the audio as either real or fake. To support interpretability, we generated visual outputs showing both the waveform and the Mel-spectrogram for each sample. These visualizations help illustrate the model's response and provide insight into the features influencing its decisions.

We also generated additional fake audio samples using various online voice synthesis tools to test the system's robustness. Most of these manually crafted deepfakes were correctly classified. However, one sample from ElevenLabs was misclassified as real, showcasing the limitations of the model when facing highly realistic synthetic voices. Table IV summarizes these results.

TABLE IV
MODEL PERFORMANCE ON MANUALLY GENERATED DEEPFAKE AUDIO SAMPLES

Generation Tool	Input Type	Ground Truth	Predicted Label
FakeYou	Voice Clone	Fake	Fake
ElevenLabs	Text-to-Speech (TTS)	Fake	Real
ElevenLabs	Text-to-Speech (TTS)	Fake	Fake
ElevenLabs	Text-to-Speech (TTS)	Fake	Fake
AI Voice	Text-to-Speech	Fake	Fake
FakeYou	Voice-to-Voice	Fake	Fake

To demonstrate the system's practical applicability, the trained model was also integrated with a frontend interface to simulate a real-world usage scenario. Although the system has not been deployed to a public server, it operates as a complete end-to-end pipeline in a locally hosted environment. Users can upload audio files via the interface and receive classification results in real time, along with visual feedback.

An example of this prediction output is shown in Figure 3, where the model's decision is displayed above the waveform and spectrogram. This allows for an intuitive understanding of how different types of audio are handled by the system.

VI. ZERO TRUST SECURITY INTEGRATION

Our approach uses Zero Trust Security (ZTS) concepts to protect both the model and user data given the sensitive character of voice data and the growing complexity of deepfake generation methods. The architecture follows the "never trust, always verify" philosophy. To enforce data confidentiality and integrity, all uploaded audio files are encrypted using the Advanced Encryption Standard (AES) before storage or transmission. This ensures that the original waveform data is never exposed in its raw form and remains secure even in case of interception. Additionally, the trained DeepSonarCNN model is protected through Fernet symmetric encryption, preventing unauthorized access or tampering with model weights.

VII. LIMITATIONS

Despite achieving 97% accuracy on the benchmark dataset, the system has some limitations. A few deepfake samples, particularly those made using powerful vocal synthesis methods, were misclassified as real, as seen in Table IV. This highlights the need for greater generalisation against highly realistic synthetic voices. Although AES encryption secures audio data in transit and storage, and Fernet encryption protects the trained model from unauthorised access or modification, both measures do not protect against adversarial audio inputs aimed to alter predictions during inference time. Furthermore, the system currently relies on GPU-based processing for best performance, which may limit its use on edge devices or in low-resource contexts. Future enhancements could include adversarial training techniques, model compression, and real-time resilience testing in noisy or distorted environments.

VIII. CONCLUSION

In this paper, we introduced DeepSonarCNN, a convolutional neural network-based model for deepfake audio detection using time-frequency analysis. Utilizing Mel-spectrogram representations of speech, the model successfully extracts fine patterns and inconsistencies that separate real speech from synthetically produced audio. The model trained and tested on the extensive Fake-or-Real (FoR) dataset, DeepSonarCNN attained excellent classification performance with a 97% accuracy rate. The robustness of the system was also confirmed by visual inspection of predictions on unseen audio, utilizing waveform and spectrogram output to further increase model transparency. In addition to detection accuracy, incorporation of Zero Trust Security principles like AES and Fernet encryption, it guarantees end-to-end security for the data and model pipeline.

REFERENCES

[1] M. Mahmud, T. Noor, and R. Hossain. "Audio Deep Fake Detection with Sonic Sleuth Model". In: *Computers* 13.10 (2024), pp. 1–16. DOI: 10 . 3390 / computers13100256.

[2] J. Wang, H. Li, and M. Chen. "MFAAN: Unveiling Audio Deepfakes with a Multi-Feature Authenticity Network". In: *arXiv preprint* (2024). arXiv: 2311.03509.

[3] K. Shaik and S. K. Shaik. "Deepfake Audio Detection via MFCC Features Using Machine Learning". In: *International Journal of Advanced Computer Science and Applications* 14.1 (2023), pp. 123–128.

[4] C. Yi, S. Yang, and M. Liu. "Audio Deepfake Detection: A Survey". In: *arXiv preprint* (2023). arXiv: 2308.14970.

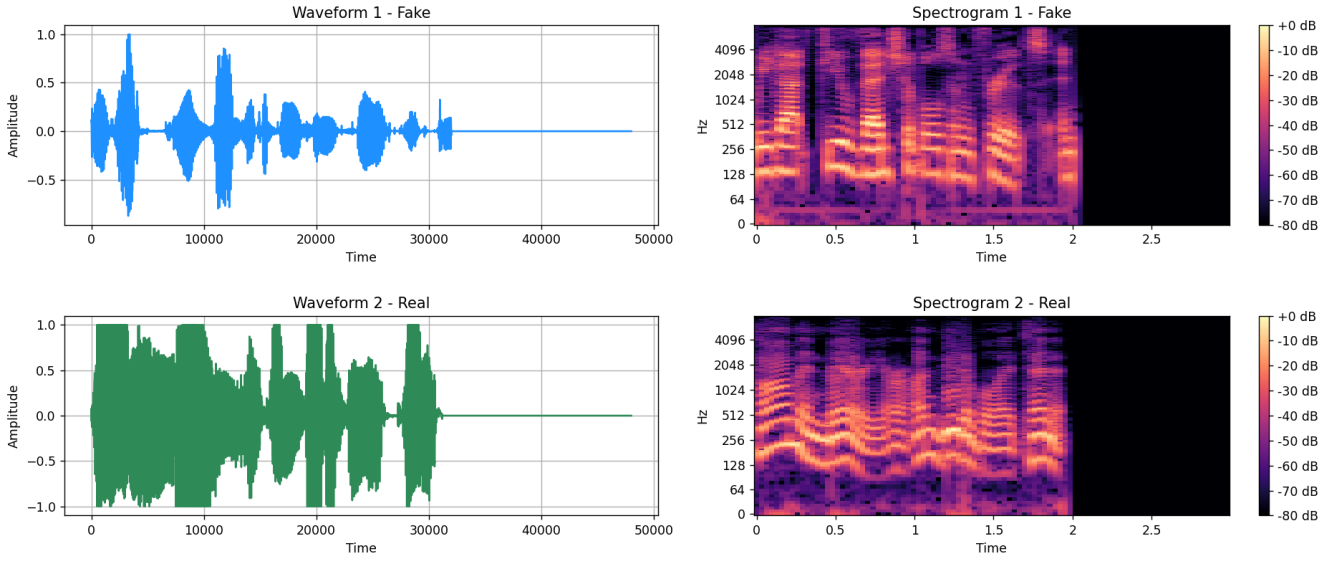

Fig. 3. Predicted waveform and spectrogram of unseen Fake and Real audio samples

[5] Ricardo Reimao and Vassilios Tzerpos. "FoR: A Dataset for Synthetic Speech Detection". In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE. 2019, pp. 3921–3926. DOI: 10.1109/SMC.2019.8914257.