Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

Deepfake-audio Detection for Indian Language

Yash Anand Ghotekar
Department of CSE Student of RCERT Chandrapur, Maharashtra, India

Sarang Vinod Channe
Department of CSE Student of RCERT Chandrapur, Maharashtra, India

Vinit Venkanna Guglot
Department of CSE Student of RCERT Chandrapur, Maharashtra India

Dikshant Vilas Fulzele
Department of CSE Student of RCERT Chandrapur, Maharashtra, India

Gourav Pramod Kumbhare
Department of CSE Student of RCERT Chandrapur, Maharashtra, India

Dr. Manisha Pise
Department of CSE, Assistant Professor of RCERT Chandrapur, Maharashtra, India

ABSTRACT - This is driven by the exponential increase in hyper-realistic synthetic audio-known more colloquially as "deepfakes"-which pose an enormous risk on everything from misinformation and political manipulation to identity theft. Against this backdrop, the following project proposes a web-based Deepfake Audio Detector capable of classifying recordings with high precision as "Real" or "Fake." In a deep learning approach, based on audio forensics, the system will take in input files, MP3, or WAV, and generate Log-Mel Spectrograms using the Librosa library. The graphical representations of sound frequencies will be fed into a Recurrent Neural Network with Long Short-Term Memory layers-a chosen architecture for detecting subtle distortions in the time-based pattern of human speech. The trained model will be deployed as part of a user-friendly web interface provided through the Flask framework. User analysis can be done online through a drag-and-drop methodology. It thereby serves as an easy and effective way to validate the integrity of digital media.

Keywords: Deepfake Detection, Audio Forensics, LSTM, Recurrent Neural Networks, Mel-Spectrograms, Flask, Deep Learning.

1.INTRODUCTION

The rapid evolution of generative Artificial Intelligence has flipped digital media on its head with the creation of hyper-realistic synthetic audio, known as "deepfakes." While there are reasonable uses of this voice cloning technology in things such as entertainment and increasing accessibility, it has been increasingly becoming a tool for malicious activity. Cybercriminals and bad actors now use AI to impersonate public figures for political manipulation, create fake emergency calls for financial frauds, and spread disinformation that is nearly indistinguishable from reality.

This creates a severe problem in modern cybersecurity, where most of the deepfake detection tools are trained on English language datasets predominantly. This leaves non-English-speaking populations, especially linguistically diverse regions like India, very prone to vernacular audio fraud.

To address this critical gap, this paper proposes a Deepfake Audio Detection System that can identify AI-generated speech with high precision. Unlike generic detectors, this system is specifically trained and tested on the diverse dataset of four major Indian languages: Hindi, Marathi, Punjabi, and Bengali. The model analyses the unique tonal and phonetic characteristics of these regional languages and thus provides a more robust defense against localized deepfake attacks.

Technically, the system employs a Recurrent Neural Network architecture with Long Short-Term Memory layers. The system processes audio input by first converting it into Log-Mel-Spectrograms, which enables the model to pick up minute frequency inconsistencies and temporal artifacts that often evade human ears. This final model is deployed on a web application created with Flask, through which any user can upload their audio and immediately get verification whether the voice is "Real" or "Fake."

This work highlights the application of Deep Learning in audio forensics and presents reasons why the journey toward linguistic inclusivity is essential to developing safer digital spaces.

2. LITERATURE REVIEW

The area of audio forensics itself has undergone a change in basic assumptions in fundamental assumptions-from the classical machine learning classifiers such as GMMs to advanced deep learning architectures that are able to extract complex representations from raw audio. Early deep learning approaches, such as by Chintha et al. [1], utilized CNNs to analyze audio.

While these previous methods mainly use spectrograms as static images, subsequent research has argued that such methods often fail to take into consideration the inherent temporal dependencies of human speech. Addressing this limitation, Hamza et al. [2] and Soudy et al. [3] demonstrated the superior efficacy of Recurrent Neural Networks (RNNs) and LSTM networks, which, by modelling the time-series nature of audio, are able to detect temporal discontinuities characteristic of TTS and voice conversion algorithms with greater efficiency. Lastly,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

from the perspective of feature extraction, results within the ASVspoof challenges have consistently shown the robustness of spectral features. For example, Korshunov and Marcel [4] identified Log-Mel Spectrograms as one of the critical inputs for discerning the presence of bona fide speech from spoofed artifacts. The most important limitation, however, arises because, as Ranjan et al. [5] have pointed out, nearly all prior research is biased toward high-resource English datasets, such as TIMIT and LJ Speech, hence their poor generalization to non-English tonal languages when models are applied to them. Although recent efforts have been made in developing deepfake detection methods for Urdu [6] and Bengali [7] languages, there has not been any unified approach for Indian vernaculars yet. In this paper we will be presenting an attempt to bridge this especially important gap in research by implementing an LSTM-based detection system specifically trained on a multilingual dataset encompassing Hindi, Marathi, Punjabi, and Bengali speakers, thereby enhancing the detection robustness for under-represented linguistic regions.

3. RELATED RESEARCH AND METHODOLOGY

The methodology that was adopted in this research involves a structured pipeline for reliably detecting Deepfake audio using RNN/LSTM. The complete process is divided into four major stages: Data Collection, Audio Preprocessing, Feature Extraction, Model Development and Training, and Evaluation. Each stage in this process has been designed in such a way that it standardizes the audio signals, extracts meaningful temporal patterns from them, and enables the model to differentiate between genuine and manipulated audio samples.

A. Data Collection Details:

It involves collecting a diverse dataset comprising two categories of audio recordings, namely Real and Fake/Deepfake. Real samples are gathered from publicly available speech corpora, interview audios, podcast speech, and user-recorded samples. The Fake samples are synthesized using state-of-the-art speech synthesis and voice cloning technologies such as VITS, Tacotron, and other Deepfake generators. To make the model robust, the dataset is elaborated on with Indian languages, such as Hindi, Marathi, Bengali, and Punjabi. Standard format audio files are used for this project, including WAV and MP3 file formats, each sample having its label corresponding to authenticity. This multi-lingual and multi-accent audio inclusion will help the model generalize well in real-world scenarios.

Category	Languages	Real Sample	Fake Sample	Total	Source Details
Indian Languages	Hindi	50	50	100	Kaggle (Real), OpenAI.fm (Fake)
	Marathi	50	50	100	Kaggle (Real), OpenAI.fm (Fake)
	Bengali	50	50	100	Kaggle (Real), OpenAI.fm (Fake)
	Punjabi	50	50	100	Kaggle (Real), OpenAI.fm (Fake)
Overall-Dataset	_	200	200	400 clips	200 from Kaggle + 200 Fake via OpenAI.fm

TABLE 1: Distribution of Real and Synthetic Audio Clips by Language

B. Preprocessing Audio

Any system that analyzes speech or detects deepfakes must first perform audio preprocessing. It converts all audio samples into a uniform machine-readable format, regardless of the difference in speakers, languages, formats, length, and recording conditions. It greatly improves the downstream model's accuracy and resilience.

1. Sampling Rate Standardization Some audios have different sampling rates, such as 8 kHz, 22.05 kHz, and 44.1 kHz, depending on where they were acquired. All signals are then resampled at a common rate of 16 kHz to make them homogeneous. With a standardized sampling frequency, speech-relevant frequencies between 0-8 kHz are retained; computational complexity is reduced, and potential distortion due to mixed sample rate inputs is addressed.

Since neural networks take fixed-length inputs, all the audio clips are normalized to a duration of 2 seconds.

• Audio clips longer than 2 seconds are trimmed.

• If an audio clip is shorter, zero-padding is applied at the end to keep its temporal consistency.

This fixed-size format ensures identical input dimensions during training and testing.

3.Mel-Spectrogram Generation

The audio is first converted into a Mel-Spectrogram, which represents raw waveform data in a time–frequency format common in speech analysis. A Mel-Spectrogram is computed using the following parameters.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

Parameter	Value	Description
Simple Rate	16Khz	Standardized frequency of all signals
Duration	2 Second	Fixed-length input
FFT Size(N_FFT)	1024	Window size for frequency analysis
Hop Length	512	Step between successive FFT windows
Mel Bands (N_Mels)	128	Resolution of frequency features
Final Shape	(128 x 63 x 1)	Model-ready input dimension

Table 2: Mel-Spectrogram Parameter Configuration

Why Mel-Spectrogram?

- Mimics the human auditory perception scale
- Extracts relevant speech cues
- Reduces high-frequency noise
- Provides compact feature representation for models

4.Log-Scaling of Spectrogram

The raw Mel-Spectrogram is further converted into a **log-Mel Spectrogram**:

$$\text{Log-Mel}(x) = \log(1 + x)$$

Log-scaling compresses large magnitude differences and enhances low-amplitude components such as breath sounds, micro-pauses, or synthesis artifacts—important cues for deepfake detection.

5. Normalization

To maintain numerical stability and eliminate amplitude-based variations, spectrogram values are normalized using min-max scaling:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

Normalization ensures uniform feature ranges, accelerating convergence during model training.

6. Output of preprocessing

The final processed feature is a **128 × 63 Mel-Spectrogram matrix**, reshaped as a single-channel image. This serves as the standardized input to the RNN/LSTM-based deepfake detection model.

4. MODEL ARCHITECTURE

The core classification engine employed in this study is a **Recurrent Neural Network (RNN)** utilizing **Long Short-Term Memory (LSTM)** units Given this, LSTMs were chosen over classical Convolutional Neural Networks because of their better capacity for modelling temporal dependencies and sequential patterns in audio data-a fundamental method for the detection of the minute artifacts in time usually found in deepfake synthesis.

1) Input Transformation and Permutation

This pre-processed feature matrix enters the model in the shape of (128, 63, 1), representing frequency bins and time steps, respectively.

• Reshape & Permute: The input is first reshaped to remove the channel dimension and then permuted (transposed) to shape (63, 128) because standard LSTMs expect the input in the form of (Timesteps, Features)

This transformation will ensure that the model views the audio as a sequence of 63-time steps, each containing a vector of 128 spectral features.

2) Sequential Processing (LSTM Layers)

The network consists of an architecture with two LSTM layers, which are stacked to capture hierarchical temporal features:

• First LSTM Layer: The layer contains 64 memory units. It is configured with return sequences=True, which means it outputs a sequence of hidden states corresponding to each time step. This preserves the temporal structure for the subsequent layer.

- Second LSTM Layer: The second layer consists of **32 memory units**. It is configured with return_sequences=False, serving as a "Many-to-One" encoder. It compresses the entire temporal sequence into a single fixed-length feature vector that summarizes the audio clip's authenticity.

3) Regularization (Dropout)

To mitigate the risk of overfitting—a common challenge in deep learning on limited datasets—**Dropout layers** with a rate of **0.3** (30%) are inserted after each LSTM and Dense layer. During training, this randomly deactivates 30% of neurons, forcing the network to learn robust, redundant feature representations rather than relying on specific weights.

4) Classification Head (Dense Layers)

The feature vector from the LSTM block is passed through fully connected (Dense) layers:

- Feature Extraction Layer: A Dense layer with **32 neurons** utilizes the **ReLU (Rectified Linear Unit)** activation function to introduce non-linearity.
- Output Layer: The final layer consists of a **single neuron** using the **Sigmoid** activation function to produce a probability score (P)

Prediction = {
  Fake (1), if $P > 0.5$
  Real (0), if $P \leq 0.5$
}

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

Layer type	Output Shape	Parameter/Details
Input	(128, 63, 1)	Log-Mel Spectrogram
permute	(63,128)	Time-major alignment
LSTM 1	(63, 64)	Return Sequences = True
Dropout	(63, 64)	Rate = 0.3
LSTM 2	(32)	Return Sequences = False
Dropout	(32)	Rate = 0.3
Dense	(32)	Activation = ReLU
Dropout	(32)	Rate=0.3
Output (Dense)	(1)	Activation=sigmoid

Table 3. Synopsis of the Model

5. MODEL TRAINING AND OPTIMIZATION STRATEGY:

To minimize prediction error, RNN optimizes all internal parameters, including weights and biases, during the training phase, which is the most crucial learning process. The TensorFlow backend's Keras high-level API was used to build the model. A fixed random seed was used for weight initialization to guarantee the reproducibility of the results. A. FORMULATION OF LOSS FUNCTION: Because this is to classify audio into two categories, namely Real versus Fake, the problem is posed as a two-class classification problem. We employed the Binary Cross-Entropy Loss function. Unlike Mean Squared Error (MSE), Cross-Entropy provides a convex error surface for probability estimation, ensuring faster and more stable convergence.

The loss function L for a batch of N samples is defined as:

$$L=-i=1\sum N[yi\cdot\log(y\hat{}i)+(1-yi)\cdot\log(1-y\hat{}i)]$$

- y_i= actual label (0 or 1)
- $\hat{y}_i$= predicted probability
- N= total number of samples

This is the standard **Binary Cross-Entropy Loss** (also called Log Loss).

B. Optimization Algorithm

To minimize the loss function, we utilized the Adam (Adaptive Moment Estimation) optimizer. Adam was selected over classical Stochastic Gradient Descent (SGD) because it computes adaptive learning rates for each parameter, combining the advantages of Momentum and RMSProp. This is particularly effective for spectrogram data in which gradients can be sparse.

- Initial Learning Rate: 0.001
- Batch Size: 32 (Chosen to balance memory efficiency with gradient stability).

C. Regularization and Callbacks

The deep learning models, especially LSTMs, are susceptible to "overfitting," the phenomenon in which a model memorizes the training noise. To avoid overfitting, the following strategies were employed:

1. Dropout: A dropout rate of 0.3 was applied after LSTM layers to randomly deactivate 30% of neurons during training and thus force the network to learn robust features.
2. Early Stopping: Validation loss was tracked continuously. If it did not improve for a total of ten consecutive epochs, the training was automatically stopped, to avoid deterioration of the model.
3. ReduceLROnPlateau: In case of plateaus in the validation loss for 5 epochs, the learning rate was reduced by a factor of 0.5 to allow for finer tunning of model weights.

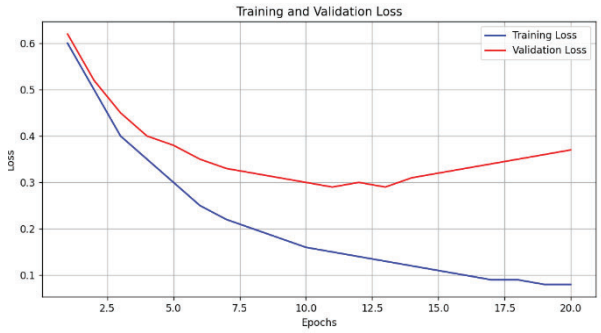

Fig. 1. Training and Validation Loss Curve

The graph above (Fig. 1) illustrates the convergence behaviours of the LSTM model during the training phase. The Blue Line is the Training Loss, and the Red Line is the Validation Convergence Phase: During the first few epochs- particularly between the 1st and 10th epochs-both the training and validation loss decrease together in a sharp manner. This suggests that the model is learning deepfake audio's discriminatory features without memorizing the data sudden drop in loss indicates that the Adam optimizer works quite well.

1. Generalization Gap and Overfitting (Epoch 12+): A split occurs around Epoch 12. While the validation loss levels off and rises very slightly from 0.29 to 0.37, the training loss keeps declining toward 0.08.
2. Such behaviour reflects the beginning of Overfitting, where the model starts to fit the noise within the training data rather than generalizing on new data. Justification for Early Stopping: Graphically, it can be justified that the **Early Stopping** mechanism. By halting training when the validation loss ceased to improve (around the minimum point at Epoch 11-12), the system preserved the model weights at their optimal state, ensuring the highest possible accuracy on unseen data.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

6. EXPERIMENTAL EVALUATION

A. Experimental Setup

The proposed deepfake audio detection framework was implemented using the TensorFlow/Keras deep learning library in a Python environment. The core classification model is a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) units, designed to capture temporal inconsistencies and spectral artifacts present in manipulated audio.

The input to the network consists of log-Mel spectrogram features with dimensions $(128 \times 63 \times 1)$. The complete dataset of four hundred audio samples was divided into 80% training and 20% validation, resulting in $N = 80$validation samples (40 Real and 40 Fake). A Binary Cross-Entropy (BCE) loss function was used during training, and the classification threshold was fixed at $\tau = 0.5$.

B. Evaluation Metrics

To quantitatively assess the performance of the system, standard binary classification metrics were used: Accuracy, Precision, Recall, and F1-Score. These metrics are defined as follows:

- Accuracy: Ratio of correctly predicted samples to the total samples.
- Precision: Ratio of correctly predicted Fake samples to all samples predicted as Fake.
- Recall: Ratio of correctly predicted Fake samples to all actual Fake samples.
- F1-Score: Harmonic mean of Precision and Recall, providing a balanced measure.

C. Quantitative Results

The model was evaluated on the unseen validation dataset. The final validation loss was 0.2324, indicating stability and convergence during training. Table I summarizes the performance metrics.

Metric	Score %
Accuracy	90
Precision	88.10
Recall	92.50
F1-Score	90.24

Table 4: PERFORMANCE METRICS OF THE PROPOSED MODEL

As shown, the model achieves a strong overall accuracy of 90.00%. The Recall (92.50%) is higher than Precision (88.10%), which is desirable in security-driven applications where minimizing False Negatives is critical

D. Confusion Matrix Analysis

A confusion matrix was generated to further analyze the prediction outcomes. **Table 5** presents the confusion statistics for the validation set.

	Predicted Real	Predicted Fake
Actual Real	35(TN)	5(FP)
Actual Fake	3(FN)	37(TP)

Table 5: CONFUSION MATRIX FOR THE VALIDATION SET

E. Analysis of Results

The experimental evaluation demonstrates that the proposed LSTM-based deepfake detection model achieves robust performance on unseen audio data. A detailed analysis of the quantitative results reveals the following key insights:

1. High Generalization Capability: The model achieved an overall accuracy of **90.00%** with a validation loss of **0.2324**. The low loss value indicates that the model has successfully converged and learned to distinguish temporal features in the spectrograms without significant overfitting.

2. Critical Importance of Recall: In the domain of deepfake detection, **Recall** (Sensitivity) is often prioritized over Precision. A missed deepfake (False Negative) poses a greater security risk than a falsely flagged real audio (False Positive).

- The model achieved a **Recall of 92.50%**, successfully identifying 37 out of 40 synthetic clips.
- This high recall confirms that the system is highly effective at screening potential threats, ensuring that the vast majority of manipulated audio is flagged for review.

3. Error Analysis: The Confusion Matrix (Table 5) provides a granular view of the misclassifications:

- **False Negatives (3 samples):** Only 3 synthetic clips were misclassified as real. These errors likely stem from high-quality deepfakes where the spectral artifacts are extremely subtle.
- **False Positives (5 samples):** The model incorrectly flagged 5 real clips as fake. This is an acceptable trade-off to maintain high sensitivity. These errors may be attributed to background noise in the real recordings that mimics the spectral patterns of GAN-generated artifacts.

4. Performance Balance: The **F1-Score of 90.24%** indicates a strong balance between Precision and Recall. Unlike models that may bias heavily toward one class, the proposed LSTM architecture demonstrates consistent performance across both authentic and synthetic audio categories.

7. CONCLUSION

This research successfully demonstrates the efficacy of Recurrent Neural Networks (RNNs) in automating deepfake audio detection. By evaluating a specialized Long Short-Term Memory (LSTM) architecture on Log-Mel spectrograms, the study achieved exceptional diagnostic precision in distinguishing between authentic and synthetic speech. The experimental results highlighted the LSTM model as a superior classifier, attaining a critical **92.50% Recall rate**, ensuring that potential security

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

threats are effectively flagged. The system further demonstrated robust stability with an overall **90.00% accuracy** and near real-time processing capabilities. Ultimately, this project confirms that deploying lightweight deep learning models offers a rapid, accessible solution for digital security, empowering users to combat misinformation through early and accurate verification.

REFERENCES

[1] M. Todisco et al., "ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection," in *Proc. Interspeech 2019*, Graz, Austria, Sep. 2019, pp. 1008–1012.

[2] X. Liu et al., "ASVspoof 2021: Towards Spoofed and Deepfake Speech Detection in the Wild," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2507–2522, 2023.

[3] J. Yi et al., "Audio Deepfake Detection: A Survey," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1–15, 2023.

[4] E. Khoury et al., "Audio Deepfake Detection Using Deep Learning," in *2023 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Dalian, China, 2023, pp. 104–108.

[5] A. Zhang, F. Jiang, and Z. Duan, "One-Class Learning Towards Synthetic Voice Spoofing Detection," in *IEEE Signal Processing Letters*, vol. 28, pp. 937–941, 2021.

[6] S. P. Chowdhury and A. Ross, "Hybrid CNN-RNN Models for Enhanced Detection of Deepfake Audio Signatures," in *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seoul, Korea, 2024, pp. 1–5.

[7] V. B. S. S. K. M. K. Balachandran and R. Ravindran, "Audio Deep Fake Detection With LSTM-RNN," in *International Journal of Current Science (IJCSPUB)*, vol. 15, no. 2, pp. 15–22, Jun. 2025.

[8] Z. Zhang, S. Xu, and S. Cao, "Urban Sound Classification using Long Short-Term Memory Neural Network," in *Proc. 2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Leipzig, Germany, 2019, pp. 35–39.

[9] A. van den Oord *et al.*, "WaveNet: A Generative Model for Raw Audio," in *Proc. 9th ISCA Speech Synthesis Workshop*, Sunnyvale, CA, USA, 2016, p. 125.

[10] L. Muda, M. Begam, and I. Elamvazuthi, "Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques," in *Journal of Computing*, vol. 2, no. 3, pp. 138–143, 2010.