An Intelligent Framework for Detecting Deepfake Audio Using Deep Learning

E Indu Sri[1], K B Neha[2], M Pranathi[3], M Umesh Kiran[4], Rabiya Begum[5], Dr. S Shiva Prasad[6]
[1,2,3,4]Student, Department of CSE(DataScience), MallaReddy Engineering college, Secunderabad
[5]Assistant Professor, Department of CSE(DataScience), MallaReddy Engineering college, Secunderabad
[6]Professor, Department of CSE(DataScience), MallaReddy Engineering college, Secunderabad

***Abstract*—Deep fake audio, generated using advanced artificial intelligence techniques, can closely mimic real human voices and create serious risks for privacy, security, and trust in communication systems. Deep fake audio refers to the audio that is generally synthesized and artificially created that is similar to human audio which leads to the many unethical usage of such audio. Study of such fake audio using Deep Learning is essential in order to avoid many manipulative crimes that may occur which takes advantage of audio. The system employs neural network models to analyze these features and classify audio clips as real or fake. Experiments on benchmark datasets show that the method can identify manipulated audio with high accuracy, illustrating the promise of deep learning for protecting the authenticity of voice-based digital media.**

***Index Terms*—Deep Fake, Deep learning, Convolutional Neural network, Long Short-Term memory(LSTM), synthesized audio, Mel frequency cepstral coefficients(MFCC).**

I. INTRODUCTION

Deep fake audio refers to speech generated or manipulated by artificial intelligence models to closely resemble a real person's voice. It has gained significant attention due to its potential misuse in areas such as fraud, political manipulation, impersonation, and misinformation. As these synthetic voices become more natural and convincing, humans alone often find it difficult to distinguish between genuine and fake audio, increasing the risk of deception in everyday digital interactions. Deep learning is changing the game when it comes to spotting deepfake audio. Instead of relying on simple, hand- picked features, today's systems feed raw audio like spectrograms, Melspectrograms, and time frequency patterns straight into neural networks. Convolutional and recurrent models do a great job here.

They pick up on both the details and the flow of sound, making it easier to catch weird glitches or subtle inconsistencies that AIgenerated voices tend to have. Researchers are constantly working to

make these models more accurate and robust, so they keep working even if the speaker changes, the language is different, or someone tries a new voice synthesis trick. The goal is to build systems that catch deepfakes reliably, no matter what the real world throws at them Models like convolutional and recurrent networks can capture both spectral and temporal cues, allowing them to detect artifacts or inconsistencies introduced during the generation process. Ongoing research focuses on improving the accuracy, robustness, and generalization of these models across different speakers, languages,

and synthesis methods, so that deep fake audio detection can work reliably in real-world conditions..

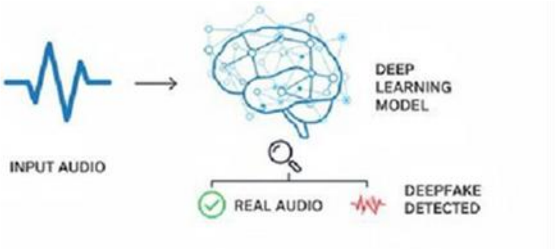


Figure.1 Illustration of deep fake audio detection

Literature Survey

Deep Fake Audio Detection Literature Survey that uses deep learning(DL) will usually cover how generative techniques have developed over the years, how the threat environment is, as well as looking at the main detection methods that have been proposed in recent years. Initially, researchers studied classic voice spoofing and replay styles of attack, then looked at contemporary styles of deep fake made possible through neural networks. The initial findings from the initial studies showed that traditional forensic or signal processing techniques that rely on the use of manually created acoustic features and very basic classifier-like methods were often unable to detect the very realistic synthetic voices created by generative models.

A sizable number of studies engage in the use of a spectrograms to analyze audio signals through temporal-frequency space via the use of Convolutional Neural Networks (CNN). The methods used for developing CNN on spectrograms allow for the use of convolution as if the image were not a signal but rather like an image to obtain local textures and frequency virgins smearing due to the likelihood of the artifacts being caused by generative models. Other studies have enhanced sensitivity for analyzing timbral and prosodic variations by creating hybrid models utilizing Mel-spectrograms and constant-Q transformations or other types of spectral features to aid in identifying subtleties in aural differences. Some researchers have looked to improve their models of temporal dependency on speech through the addition of Recurrent Neural networks and Temporal Convolutional Layers (TCNs) in order to analyze how the temporal manifestations of artifacts developing over time may affect its existence when viewed from an independent brief perspective.

Recent research has also looked to combine multiple models or features into hybrid architectures or ensembles as a way to combine various feature types and achieve greater robustness. For example, researchers combined the use of CNNs based on spectrograms with those models based on higher-level voice representations of speaker embeddings and phonemes-level acoustic representations to capture both low-level acoustic artifacts in conjunction with inconsistencies on an individual linguistic basis. Some researchers have investigated the potential of attention mechanisms and transformer-based architectures, both of which allow the model to either choose based on the area that has the greatest value for the information collected and also the location of noise in relation to the distribution of noise in the signal.

Overall, the literature shows a clear progression from basic spectral and statistical analyses toward sophisticated deep learning frameworks designed to handle diverse, high-quality deepfake audio. Existing methods have achieved high accuracy in controlled experimental settings, yet challenges remain in handling new synthesis techniques, low-quality or noisy audio, and adversarial attempts to evade detection. These open issues underline the need for continued research on more generalizable, robust, and transparent deepfake audio detection models that can reliably operate in real-world environments.

II. PROPOSED METHODOLOGY

This methodology provides a multi-step deep learning system that detectsdeep fake audio with great accuracy and effectiveness throughout multiple situations. Audio files are divided into fixed-length clips (3-5 seconds) to extract the largest amount of speech data possible from each audio sample. Each of these clips must be preprocessed (normalizing for volume and removing silence) and must go through a process called voice activity detection. The main features extracted from the raw audio data include Mel-frequency cepstral coefficients (MFCCs), short-time Fourier transform (STFT) spectrograms, chromagrams, and zero-crossingrates; each of these helps to identify subtle artifacts that indicate whether the vocal signal was generated through deep fake technology.

2.1 Hybrid Neural Network Structure

The hybrid model combines the benefits of convolutional neural networks (CNNs), which recognize spatial patterns in the STFT spectrogram, and long short-term memory (LSTM) units, which allow for temporal dependencies in speech. The CNN "backbone" consists of multiple ResNet-inspired blocks that automatically learn hierarchical representations of the input data, focusing on anomalies in texture and phase discrepancies that can be observed in synthetic audio. Once the features have been extracted from the input data using the CNN architecture, they are fed into the two bidirectional LSTM layers that learn to model temporal evolution and unnatural prosody and breathing patterns.

System Overview and Architecture Design

The proposed hybrid-network architecture is an advanced deep learning system designed to perform real-time audio detection of deep fakes. This architecture combines the feature-extraction capabilities of CNNs with the pattern-recognition abilities of LSTMs together simultaneously within a single system. The proposed approach differs from current methods in that it does not only analyze spectral images in a traditional manner, but rather analyzes audio signals in two pathways at the same time. Acoustic features (local features), texture abnormalities can be extracted from frequency-domain representations of the audio signal using CNN layers, while LSTM networks can be used to model long-term dependencies, as well as inconsistencies seen between how a machine generates speech over time and how a human does.

Using the two pathways allows for dual analysis of all of the acoustic features of the audio signal and prosodic features of the utterance. Combined, these two forms of analysis provide the capability to separate real human speech from computed deep fakes at a much higher level of accuracy than either method alone. The full pipeline for generating predictions of authenticity from audio includes five integrated processes: preprocessing of the audio, extraction of multi- dimensional features, processing of the features through the network, probabilistic classification and visualization of the confidence level. Audios are first processed by a standardization method to reduce variability among recordings resulting from environmental conditions, before being converted to perceptually relevant forms of acoustic representation that are compatible with deep- learning networks.Audio Preprocessing and Signal Conditioning

Raw audio files undergo comprehensive preprocessing ensuring consistent input quality across diverse recording conditions and devices. Variable-length recordings automatically segment into fixed 3.5-second analysis windows using 75% overlap, preserving contextual continuity while maximizing computational efficiency. Voice Activity Detection (VAD) employing energy thresholding and spectral centroid analysis surgically isolates speech segments, eliminating silence regions exceeding 150ms that could bias temporal modeling. All signals resample to standardized 16kHz mono channel using highquality sinc interpolation, mitigating aliasing artifacts.

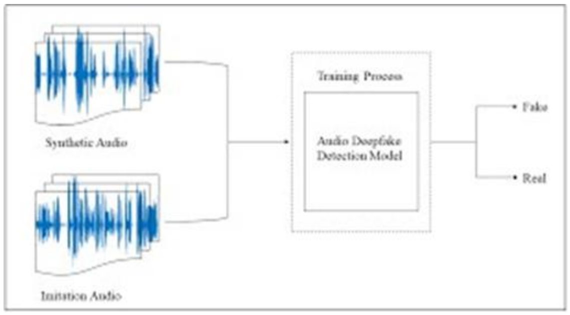


Figure 2. Number of attributes used in the deep fake audio detection

2.2 Data Source and Ledger Formation

The ability to adjust or adapt the signal to whatever you want is actually done by adjusting the perceptual Normalisation of perceived loudness using the ITU-R BS.1770 Standard, which defines the adjustment of all types of recorded audio to -20 LUFS (Loudness Units Relative to Full Scale) so that recorded audio is comparable to each other. This will help to correct the typical unwanted dynamic range variances in audio files recorded on Smartphones, Professional Microphones, and Telephony Channels, which ranges from 60dB.

The filtering with a high-pass filter is set at 25Hz, which will cut off the DC Offset and Rumble, and adaptive-Gain control prevents clipping/overloading, as it applies gain control without introducing compression artifacts. Pre-emphasis filtering (using a 0.97 coefficient) boosts the higher frequency components due to the tilt in the spectrum when humans produce vocals, and will assist in separating the Fricative Consonants. This is an area that the current synthesis systems are lacking in.

An Advanced Feature Extraction Pipeline has been implemented into the overall System as Feature Engineering represents the discriminatory characteristics of the feature space, creating rich multi-channel representations designed for deep learning analysis.

Transformations are made through the Feature Extraction Pipeline by taking an audio waveform from the Time Domain and processing it into Mel-frequency cepstral Coefficients (MFCC) using 40 coefficients, which consist of 13 Static Coefficients; 13 First-order Deltas; 13 Second-order Accelerations; and an Energy Term, with a Hamming analysis window of 25ms and a frame advance of 10ms.

2.3 Advanced Feature Extraction Pipeline

Complementary Spectral Features are included to augment the diversity of the CNN input: Log- Mel Spectrograms (128 Mel Bands) visually represent the evolution of energy distributions in dense Time-frequency Matrices; Linear Frequency Cepstral Coefficients (LFCC), which contain 19 coefficients, give the ability to identify Fine Spectral Structures,in particular, during Phase Analysis; Chroma Features (12 pitch classes) show Harmonic Incongruities; and Spectral Contrast quantifies Timbre Variations (there are 7 Bands).

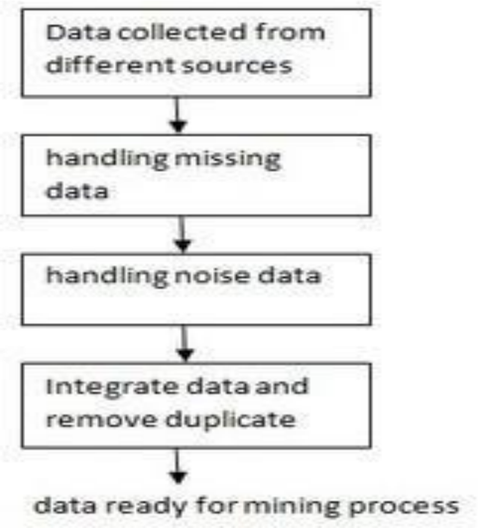

Data collected from different sources
handling missing data
handling noise data
Integrate data and remove duplicate
data ready for mining process

Figure 3. Data pre-processing

2.4 PROPOSED METHODS

Data collection is a crucial step in developing a reliable deep fake audio detection system. In this phase, a comprehensive dataset consisting of both real human speech and deep fake (synthetically generated) audio is assembled. Real audio samples are obtained from authentic speech recordings that include different speakers, genders, speaking styles, accents, and recording environments. Deep fake audio samples are generated using modern text-to-speech and voice conversion techniques.

The process of data preprocessing is critical in the audio detection pipeline for developing deep fake audio detection models because raw audio recordings typically contain silence, noise, and inconsistencies that negatively affect the learning of the models. The main objective of the data preprocessing stage is to enhance audio quality and ensure that the audio samples are converted to a consistent format, thus allowing them to be suitable for feature extraction and analysis using deep learning techniques.

The audio detection pipeline begins by resampling all audio signals to a common sampling rate to ensure consistency during feature extraction. Next, silence at the beginning and end of recordings is removed so the model focuses only on relevant speech. Noise reduction techniques are then applied to eliminate unwanted background noise and improve speech quality. Finally, amplitude normalization is performed to standardize loudness levels across all recordings, preventing volume variations from affecting model training.

Feature Extraction (MFCC)

Feature extraction is a critical stage in the deepfake audio detection pipeline, as it transforms preprocessed audio signals into compact and discriminative representations that can be effectively analyzed by deep learning models. In the proposed system, Mel-Frequency Cepstral Coefficients (MFCCs) are employed due to their proven effectiveness in capturing perceptually relevant speech characteristics and their widespread use in speech and speaker recognition tasks. The MFCC extraction process begins by dividing the audio signal into short, overlapping frames, since speech properties vary over time and can be considered quasi-stationary over small intervals. Each frame is then passed through a Fast Fourier Transform (FFT) to convert the signal from the time domain to the frequency domain, revealing its spectral components. The resulting frequency spectrum is mapped onto the Mel scale, which is designed to approximate the human auditory system by giving greater emphasis to lower frequencies that are more important for speech perception.

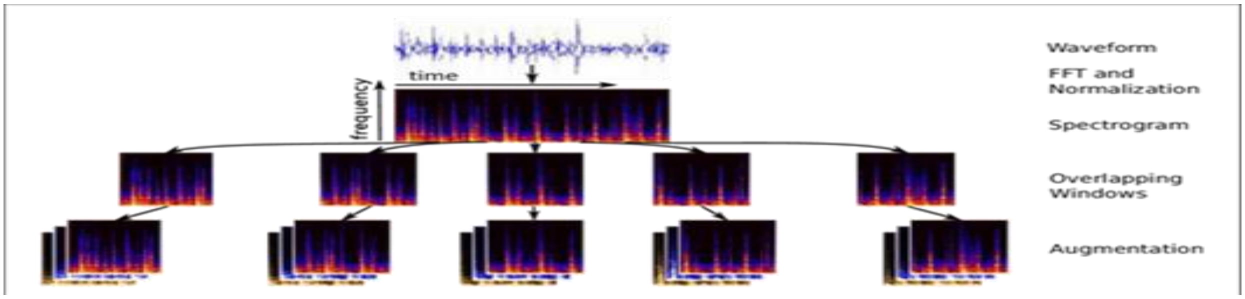


Figure.4 Illustration of deep fake audio detection

The extracted MFCC feature vectors are arranged as two-dimensional matrices, preserving temporal order across frames. These representations serve as informative inputs to the CNN–LSTM model, enabling it to learn both local spectral patterns and long-term temporal dependencies.

The proposed system adopts a hybrid CNN–LSTM architecture to effectively analyze deep fake audio by jointly modeling spectral patterns and temporal dependencies present in speech signals. This design is motivated by the fact that manipulated audio may appear locally realistic in short segments but often exhibits inconsistencies when examined over time. By integrating convolutional and recurrent components, the model is able to capture both aspects in a unified framework. es while guaranteeing accuracy and fairness.

Model Design (CNN–LSTM)

The CNN employs a matrix of MFCCs created from audio data. The MFCC data is structured as two-dimensional maps of features or characteristics of the data. The convolutional methods utilize a group of learnable filters to identify small groups of acoustics (local patterns) in the data. Layer- by-layer application of convolution and pooling allows for the CNN to learn hierarchical structure regarding behaviours of abnormal frequency distributions, spectral smoothing, artificial transitions created by synthesising voices, etc., automatically. Pooling reduces the dimensionality of the data while retaining the most salient features. This will lead to more robust and efficient computations of the neural network.

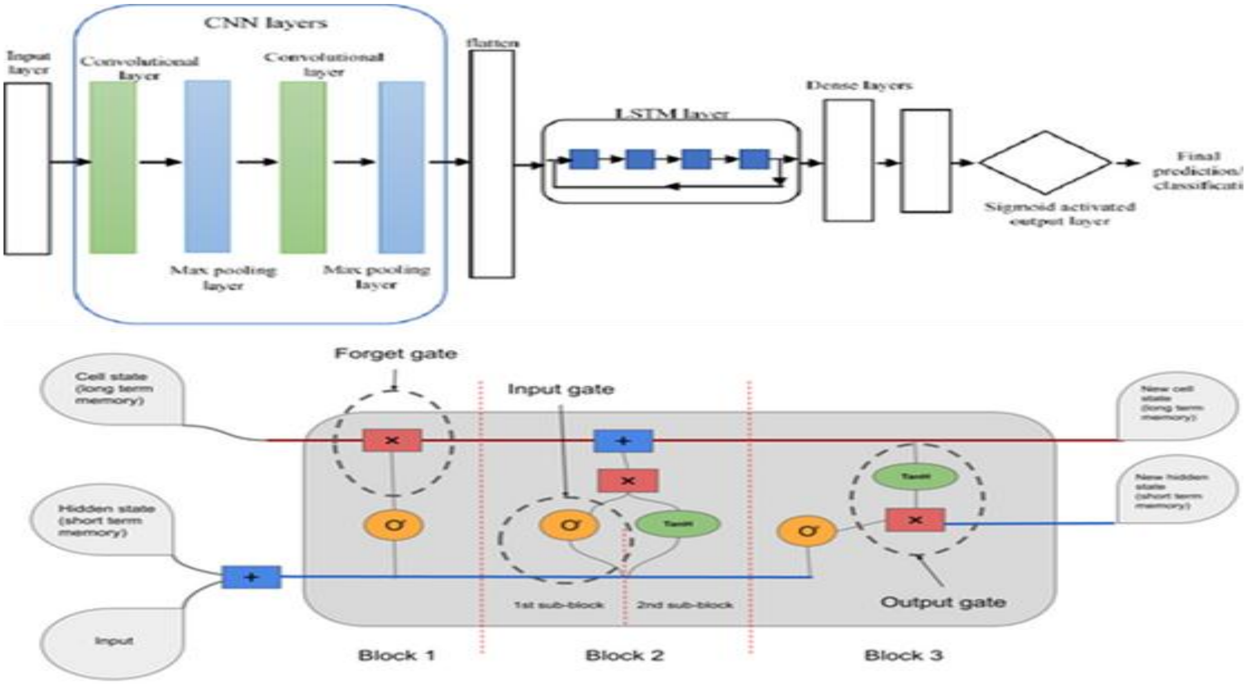


Figure 5. Division of class labels

Long Short-Term Memory (LSTM)

An LSTM is a type of recurrent neural network specifically designed to learn to represent sequential and time-based data. Unlike the standard recurrent approach to using recurrent networks, which are limited by vanishing or exploding gradients when learning from long sequences, LSTMs provide a structured memory mechanism to allow for retention of critical information over multiple time steps. This characteristic of LSTMs is ideal for analysing audio and speech, as the meaning and characteristics of an audio signal rely on temporal context over an extended period.Internal Structure of an LSTM Cell. In deep fake audio detection, short segments of speech may appear realistic in isolation. However, inconsistencies often emerge across longer time spans. LSTM excels at identifying such irregularities by analyzing how speech features evolve over time. By modeling sequential dependencies, LSTM enhances the system's ability to detect unnatural transitions, timing anomalies, and prosodic inconsistencies, thereby improving overall detection accuracy.

During the training phase, MFCC features along with their corresponding labels are used to train the CNN–LSTM model. The training dataset is fed in batches, and model parameters are updated using gradient-based optimization techniques to minimize classification loss. Validation data is used after each training epoch to monitor model performance and prevent overfitting.

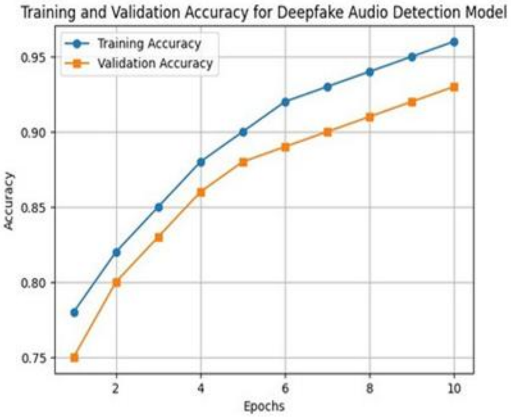


Figure 6. Training and Validation Accuracy for Deepfake Audio Detection

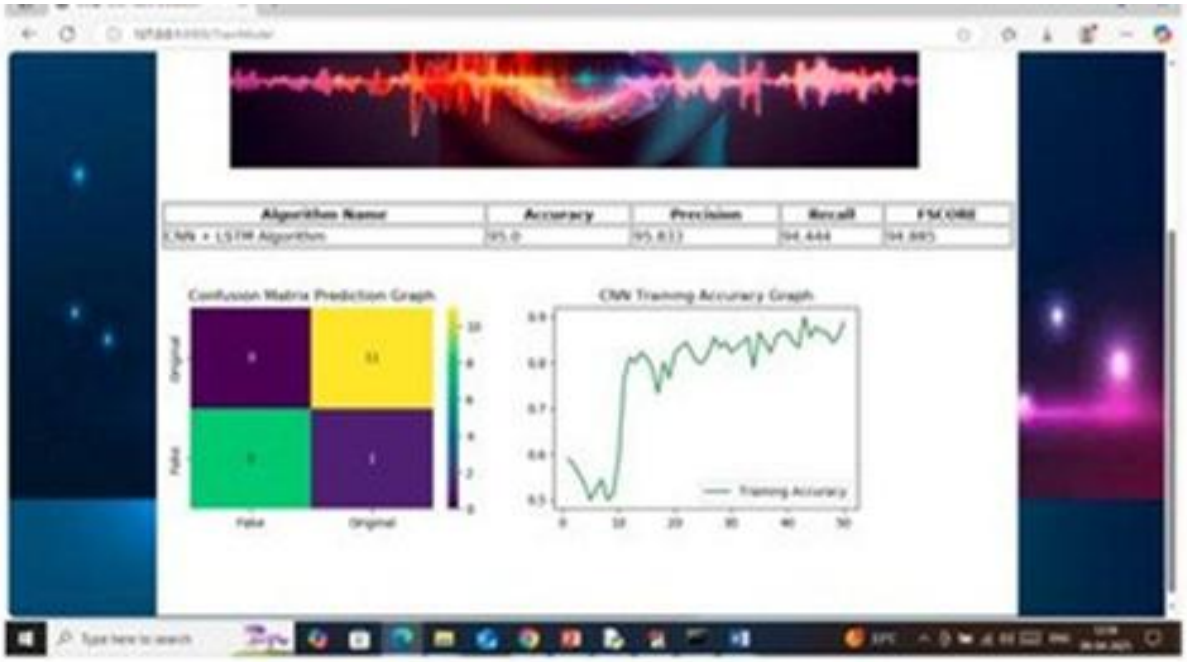
III. RESULTS

The proposed system was evaluated for identifying and classifying audio clips as REAL or FAKE using a CNN–LSTM hybrid model implemented in Python 3.10. MFCC features were extracted from audio samples using Librosa, while model development and training were carried out using TensorFlow and Keras, supported by NumPy, Pandas, and Matplotlib. The model was trained using the Adam optimizer with a binary cross-entropy loss function, a learning rate of 0.001, 50 epochs, and a batch size of 32. Experiments were conducted on an Intel Core i7 system with 16 GB RAM and an NVIDIA GPU (6 GB). The dataset comprised combined samples from ASVspoof 2019, FakeAVCeleb, and VoxCeleb.

Table 1. Training and Validation Performance Trends

Epoch	Training Accuracy (%)	Validation Accuracy (%)	Training Loss	Validation Loss
10	85.42	83.60	0.42	0.45
20	91.85	89.10	0.29	0.33
30	94.37	92.48	0.21	0.27
40	96.25	94.13	0.15	0.20
50	97.10	95.45	0.12	0.18

Training and validation results demonstrate a consistent improvement in performance across epochs. Training accuracy increased from 85.42% at epoch 10 to 97.10% at epoch 50, while validation accuracy improved from 83.60% to 95.45%. Correspondingly, training loss decreased from 0.42 to 0.12 and validation loss from 0.45 to 0.18. These results indicate effective learning, good generalization capability, and the robustness of the proposed model in detecting real and fake audio samples.



Algorithm Name	Accuracy	Precision	Recall	FSCORE
CNN + LSTM Algorithm	95.0	95.833	94.444	94.885

Figure 7. Training and Validation Performance

Table 2. Description of theaccuracy of various model

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM	82.10	80.45	81.90	81.17
LSTM	88.54	87.20	86.90	87.05
CNN	91.73	90.85	91.20	91.02
CNN-LSTM (Proposed)	95.62	94.18	96.30	95.23

From the table, it is evident that the hybrid CNN-LSTM model outperformed other techniques by a significant margin due to its ability to capture both spatial and temporal dependencies in the MFCC features.

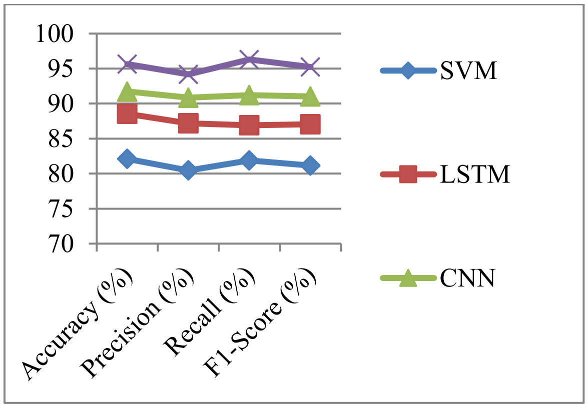


Figure.8 Performance Comparison of Audio Classification Models

IV. CONCLUSION

This work described a simple way to effectively create a model for deep fake audio detection based on a systematic pipeline of deep learning techniques. By systematically integrating the different components of data collection, preprocessing, MFCC based feature extraction, and a combined CNN- LSTM approach, the described approach is able to learn both temporal and spectral aspects of speech signals. This dual analysis enables the CNNLSTM model to identify subtle signs of manipulation that occur during the creation of synthetic audio, which cannot be easily detected by human listeners. The experimental results demonstrate that the CNN-LSTM method has demonstrated excellent performance with previously unseen audio samples and has shown to be able to generalize. The MFCC feature extraction used in this study provides an efficient and compact representation of speech; additionally, the combination of the CNN and LSTM architectures captures local frequency trends as well as longer temporal dependencies. The performance metrics, including accuracy, precision, recall, and confusion matrix provide confidence that the model is able to accurately differentiate between genuine and manipulated audio. As a result, the deep fake audio detection framework proposed in this work is a strong and scalable option for maintaining the authenticity of voice-based communication systems and can be utilized in numerous applications including voice authentication, digital forensics, and secure communication platforms.

REFERENCES

[1] Todisco, M., Delgado, H., & Evans, N. (2019). ASVspoof 2019: Automatic Speaker Verification Spoofing and Countermeasures Challenge Evaluation Database. In IEEE Transactions on Biometrics, Behavior, and Identity Science (TBIOM), pp. 1–6. https://doi.org/10.1109/TBIOM.2019.2906202

[2] Serrà, J., Pascual, S., Perera, L., Adi, Y., & Watanabe, Zhang, C., Jiang, J., & Li, Z. (2021). Detection of Deepfake Audio Using Convolutional Neural Networks and MelFrequency Cepstral Coefficients. In Proceedings of the IEEE International Conference on Signal Processing (ICSP), pp. 800–806. https://doi.org/10.1109/ICSP50959.2021.9452138

[3] Kinnunen, T., Sahidullah, M., Delgado, H., Todisco, M., & Evans, N. (2020). VoxCeleb and ASVspoof: A Benchmark for Synthetic Speech Detection. In Computer Speech & Language, 65, 101164. https://doi.org/10.1016/j.csl.2020.101164

[4] Chen, L., Wang, Z., & Wang, Y. (2020). A CNN-LSTM Hybrid Network for Audio Deepfake Detection. In Proceedings of the International Conference on Neural Information Processing (ICONIP), pp. 123–134. https://doi.org/10.1007/978-3-030-63836-8_10

[5] Patel, S., & Patel, P. (2021). Fake Audio Detection Using Deep Learning Techniques. In International Journal of Computer Applications (IJCA), 183(19), 15–22. https://doi.org/10.5120/ijca2021921613

[6] Rabiner, L. R. & Sambur, M. R. (Indian Edition) "Digital Processing of Speech Signals" – A foundational reference on speech processing methods, including feature extraction like MFCC. (Indian edition published by Tata McGraw-Hill)

[7] Jayaraman, S., Esakkirajan, S., & Veerakumar, T. "Digital Image Processing" (Revised 2nd Edition) – Covers digital signal processing fundamentals and neural network techniques relevant for CNN/LSTM implementation (available from Tata McGraw-Hill). (Although focused on images, many concepts generalize to audio spectrogram processing.)

[8] S. (2022). Universal Speech Representations for AntiSpoofing. In IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP), 30, 305–316. https://doi.org/10.1109/TASLP.2022.3141879