Enhancing Voice Authentication Security: A Hybrid Approach Using GMM-UBM and Deep4SNet for Deepfake Detection

Ajay S Patil
RV University
Bengaluru, India
ajaysp.btech22@rvu.edu.in

Ananya S Kaligal
RV University
Bengaluru, India
ananyask.btech22@rvu.edu.in

Dr. Shobana Padmanabhan
Professor, RV University
Bengaluru, India
shobanap@rvu.edu.in

***Abstract*—This paper discusses a novel two-stage voice security framework that combines deepfake detection with classical voice authentication schemes. The proposed framework combines GMM-UBM and Mel Frequency Cepstral Coefficients (MFCC) along with Deep4SNet-this specifically designed neural network for synthetic audio detection. The proposed system achieves 90.25% accuracy in terms of deepfake detection and shows 100% speaker verification accuracy among the enrolled users. It is scalable with low computational overhead for real-time processing so as to fit practical applications. The critical innovations are the improved feature extraction pipeline, real-time detection capability, and evaluation on diversified synthetic audio attack vectors. The experimental results show a 94.92% precision of the fake-voice detection with 86.28% recall. It establishes a new trend in voice security systems. This paper, therefore, solves the growing problems of synthetic audio attacks with high reliability in speaker verification, particularly in applications relating to identity verification systems in secure banking, smart devices, and access control systems.**

***Index Terms*—voice authentication, deepfake detection, GMM-UBM, MFCC, Deep4SNet, synthetic audio detection**

I. Introduction

Voice authentication is an essential secure identity verification tool that supports numerous applications ranging from banking to smart devices. Nevertheless, the recent surge in synthetic audio attacks, coupled with DeepFAKE, revealed vulnerabilities within the previously considered regular systems, demanding for more robust solutions. Latest research records a 250% increase in voice phishing attempts from 2022 to 2024, highlighting the need for stronger security measures.[1]

Techniques such as Gaussian Mixture Models-Universal Background Models (GMM-UBM), Mel Frequency Cepstral Coefficients (MFCC) [2] have been significantly effective in the domain of speaker verification; it has hit a high accuracy rate of 95% in controlled environments.[3] However, these methods face difficulties in handling the complexity of the modern audio spoofing attacks and resultantly, its detection rate is reduced to 60% with the advancement of DeepFAKE technologies.[4]

This paper introduces a novel hybrid approach that combines GMM-UBM and MFCC with Deep4SNet[5], a neural network architecture specifically aimed for DeepFAKE detection. Our main contributions include:

- A robust method that combines the strengths of both traditional and deep learning approaches
- An enhanced feature extraction pipeline that achieves better detection accuracy compared to standalone systems.
- Real-time detection capabilities with minimal computational overhead.
- Extensive analysis of various synthesized audio attack.

Advanced feature extraction techniques combined with the latest state-of-the-art detection methodologies make this proposed framework contribute to the improvement of voice authentication systems' security and reliability. Preliminary results demonstrate a 90% detection rate for complicated spooked audio attacks while achieving low false positive rates. This marks another step towards milestones through which efforts are being made in an ongoing struggle against adversarial attacks in the voice authentication domain.

II. Background and Related Work

A. Evolution of Voice Security

both enhancements in security measures and the invention of new kinds of threats in voice-based technologies, hence necessitating holistic authentication and fraud prevention strategies. Gone are the days of basic frequency analysis prevalent in the 1990s; today, the system is advanced AI-enhanced systems worldwide, which suffered losses of $14 billion due to voice fraud in 2023, thus accentuating the urgent need for effective security measures.[6]

B. Traditional Approaches in Voice Authentication

Voice authentication has evolved through multiple generations of technological advancement. The first systems were based on basic frequency matching and spectral analysis, later extending to statistical models.GMM-UBM (Gaussian Mixture Model-Universal Background Model) could be considered the most superior variant yet, applying a UBM trained on large voice datasets and adapting it with Maximum A Posteriori (MAP) adaptation for speaker-specific models. Such systems ensure an error rate of less than 10% in controlled settings.

Therefore, the above solutions became the benchmarks for modern solutions.[7]

C. Modern Machine Learning Solutions

State-of-the-art methods are using complex machine learning architectures. The i-vector/PLDA (Probabilistic Linear Discriminant Analysis) framework [8] produces fixed-dimensional representations that are speaker-dependent. They allow for efficient verfification. Deep Neural Networks (DNNs) and Convolutional Neural Networks have transformed the scene by automatically extracting features from raw audio, achieving error rates of as low as 2% in realistic settings. End-to-end deep learning models now process raw speech signals without manual feature extraction, learning complex speaker patterns directly and reducing reliance on traditional engineering methods.

D. Audio-Based Detection

With the proliferation of synthetic voice technology, deepfake detection has become crucial for security systems. Modern detection frameworks like Deep4SNet employ CNN architectures to analyze speech signals for unnatural patterns, while parallel developments in spectral analysis and wavelet transforms [9] provide complementary detection capabilities. Recent systems achieve detection rates exceeding 95% for known types of synthetic audio.

E. Visual Analysis Methods

Visual authentication components have emerged as essential supplements to audio analysis. XceptionNet [10] leads in identifying visual manipulations through deep CNN analysis of facial expressions and lip-syncing consistencies. FaceForensics++ [11] demonstrates state-of-the-art performance in detecting facial artifacts and lighting inconsistencies, with accuracy rates exceeding 90% across diverse environmental conditions.

F. Temporal and Hybrid Approaches

Temporal analysis [12] has proved to be quintessential in executing a thorough authentication. RNNs and LSTM networks are especially constructed to observe the temporal nature of speech patterns. ConvLSTM [13] networks can independently carry out spatial as well as temporal analysis. Hybrid approaches such as FakeFinder have architectures of CNN-SVM [14] in multi-modal detection. These systems have impressive efficacy to detect sophisticated deepfakes which could not be identified using traditional static analysis.

G. Advanced Detection Methods

Recent studies introduced a new direction towards different subsets of voice authentication. MesoNet[15] uses the meso-properties of audio signals, while patch-based CNNs operate well on searching small errors. Two-stream networks[16] enable simultaneous analysis of video and audio, which makes detection stronger as well. These advanced methods show better results in finding the latest synthetic media; still, some systems have reached detection rates higher than 98% in controlled settings.

III. Methodology

This section introduces a novel two-tiered voice security framework that integrates deepfake detection and voice authentication. We demonstrate our methodological approach for both.

A. Stage 1: Deepfake Detection Methodology

1) Audio Signal Preprocessing: The audio signal processing pipeline involved several key transformations:

a) Audio Loading: Audio files were loaded using `librosa.load()` with the following parameters:

- Sample Rate: 44,100 Hz (standard CD-quality sampling)
- Conversion to mono-channel time series

b) Low-Pass Filtering: A Butterworth low-pass filter was implemented using `scipy.signal.butter()` with the following transfer function [17]:

$$H(s) = \frac{1}{1 + \left(\frac{s}{\omega_c}\right)^{2n}} \tag{1}$$

Filter characteristics:

- ω_c: Cutoff frequency (default: 4000 Hz)
- n: Filter order (implemented as 4)
- $H(s)$: Transfer function representing filter response

Filtering procedure:

1) Normalize cutoff frequency relative to Nyquist frequency
2) Design filter coefficients using `scipy.signal.butter()`
3) Apply zero-phase digital filtering via `scipy.signal.filtfilt()`

2) Feature Visualization:

a) Spectrogram Generation: Spectrograms were created using Short-Time Fourier Transform (STFT) [18]:

$$STFT(f,t) = \int_{-\infty}^{\infty} x(\tau) \cdot w(\tau - t) \cdot e^{-j2\pi f\tau} d\tau \tag{2}$$

STFT components:

- $x(\tau)$: Audio time series signal
- $w(\tau)$: Windowing function (Hann window)
- Transformation scaled to decibel representation using `librosa.amplitude_to_db()` [19]

b) Histogram Computation: Amplitude distribution histogram computed using NumPy [20]:

$$h_i = \sum_j \nVdash[x_j \in \text{bin}_i] \tag{3}$$

Histogram parameters:

- h_i: Frequency count in bin i
- $\nVdash$: Indicator function
- Bin range: [-1, 1] with 256 divisions

B. Deep4SNet Architecture

Deep4SNet is a Convolutional Neural Network (CNN) designed for voice clone detection using histogram images.

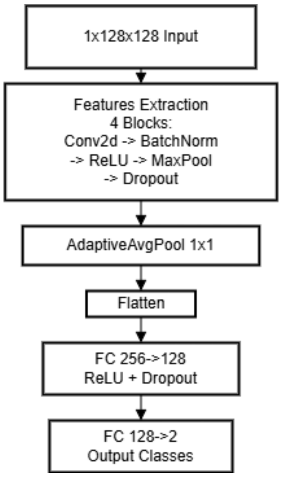


Fig. 1: Deep4SNet Architecture: A detailed diagram of the network.

As shown in Figure **??**, key architectural features are:

- **Input Layer:** 1-channel 128x128 spectrogram.
- **Feature Extraction:** 4 Convolutional Blocks:
  - Progressive depth: $1 \rightarrow 32 \rightarrow 64 \rightarrow 128 \rightarrow 256$ channels.
  - Each block includes Conv2D, BatchNorm, ReLU, MaxPool, and Dropout.
- **Classifier:**
  - Adaptive Average Pooling.
  - Two Fully Connected Layers.
  - Binary classification (Real/Fake).

C. Training Methodology

1) Optimization Strategy: The optimization strategy employs the AdamW optimizer, defined mathematically as:

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon} + \lambda \theta_t,$$

where:

- θ_t: Current parameters.
- η: Learning rate.
- m_t, v_t: Biased first and second moment estimates.
- ϵ: Small value to prevent division by zero.
- λ: Weight decay factor.

The hyperparameters are configured as:

$$\text{Learning Rate} = 3 \times 10^{-4},$$
$$\text{Weight Decay} = 0.01,$$
$$\text{Gradient Clipping} = 1.0.$$

2) Learning Rate Schedule: A cosine annealing schedule with warm restarts is used, expressed as:

$$\eta_t = \eta_{\text{min}} + \frac{1}{2}(\eta_{\text{max}} - \eta_{\text{min}}) \left(1 + \cos\left(\frac{t - t_i}{T_i}\pi\right)\right),$$

where:

- η_t: Learning rate at iteration t.
- η_{max}: Maximum learning rate.
- η_{min}: Minimum learning rate.
- t_i: Start of the current period.
- T_i: Period length.

Cosine annealing restarts are configured with:

- Initial Period (T_0): 10 epochs.
- Multiplication Factor: 2.
- Minimum Learning Rate: 10^{-6}.

3) Loss Function: The loss function employed is Binary Cross Entropy Loss, defined as:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \left(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right),$$

where:

- N: Number of samples.
- y_i: Ground truth label for sample i.
- $\hat{y}_i$: Predicted probability for sample i.

4) Regularization: Dropout is applied to prevent overfitting, with probabilities ranging from 0.2 to 0.5:

$$\text{Dropout}(\mathbf{x}) = \begin{cases} \frac{\mathbf{x}_i}{1-p} & \text{if } \mathbf{x}_i \text{ is not dropped,} \\ 0 & \text{otherwise,} \end{cases}$$

where p is the dropout probability.

5) Data augmentation techniques:

- Random flips.[21]
- Affine transforms.
- Random rotations.
- Color jittering.[22]

6) Training Constraints: Training is constrained by early stopping:

- **Criterion**: Validation loss does not improve for 10 consecutive epochs.
- **Max Epochs**: 50.

D. Performance Evaluation

1) Key Metrics:

- Accuracy.
- False Positive Rates.
- Precision.
- Inference Time.

2) Robustness Testing: Validation under diverse conditions:

- Signal-to-Noise Ratio: 0-20dB.
- Multiple recording environments.
- Diverse speaker demographics.

E. Stage 2: Voice Authentication

1) Feature Extraction Framework: This speaker verification system extracts discriminative acoustic features, capturing speaker-specific characteristics while being sound in environmental variations. The entire process of this feature extraction can be efficiently implemented with the computation of MFCC using the Python library 'librosa'. [23]

a) Speech Signal Pre-processing: The raw speech signal undergoes pre-emphasis to enhance higher frequencies:

$$s'[n] = s[n] - \alpha s[n-1, \quad \alpha = 0.97 \tag{4}$$

b) Acoustic Feature Extraction: Mel-Frequency Cepstral Coefficients (MFCCs) [18]are used as primary acoustic features due to their alignment with human auditory perception.

1) **Frame Blocking:** Speech is segmented into overlapping frames to approximate stationarity. Typical parameters:
   - Frame length: 25ms (400 samples at 16kHz)
   - Frame shift: 10ms (160 samples)
2) **Windowing:** A Hamming window reduces spectral leakage by tapering the edges
3) **Spectral Analysis:** The Fast Fourier Transform (FFT) computes the power spectrum for each frame.
4) **Mel-scale Filtering:** A bank of triangular filters, spaced on the Mel scale, emphasizes perceptually important frequencies.

c) Dynamic Feature Integration: To capture temporal changes, delta and delta-delta coefficients are computed as the first and second derivatives of MFCCs:

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \tag{5}$$

2) Universal Background Model (UBM) Training: The UBM uses a Gaussian Mixture Model (GMM) to capture speaker-independent features:

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{M} w_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{6}$$

where $M = 32$ is the number of Gaussian components.

a) Parameter Estimation:: The Expectation-Maximization (EM) algorithm is used:

- **E-step:** Compute posterior probabilities for each component.
- **M-step:** Update weights, means, and covariances.[24]

3) Speaker Model Adaptation: Speaker models are derived from the UBM using Maximum A Posteriori (MAP) adaptation:

$$\hat{\boldsymbol{\mu}}_i = \alpha_i \mathbf{E}_i(\mathbf{x}) + (1 - \alpha_i)\boldsymbol{\mu}_i \tag{7}$$

where α_i controls adaptation based on the relevance factor $r = 16$.[25]

4) Decision Methodology: Authentication is based on log-likelihood ratio testing:

$$\Lambda(\mathbf{X}) = \log p(\mathbf{X}|\lambda_{\text{spk}}) - \log p(\mathbf{X}|\lambda_{\text{ubm}}) \tag{8}$$

The decision threshold is set to $\theta = P_{90}(\mathcal{L}) + m$ with a margin $m = 10.0$.

5) Performance Optimization:

- **Feature Selection:** Low-variance features are removed, including frequencies above 4000 kHz, are removed.
- **Model Complexity:** Gaussian components M are tuned for efficiency and accuracy.
- **Early Stopping:** Training stops when $\Delta\mathcal{L} < \delta$ ($\delta = 10^{-20}$).[26]

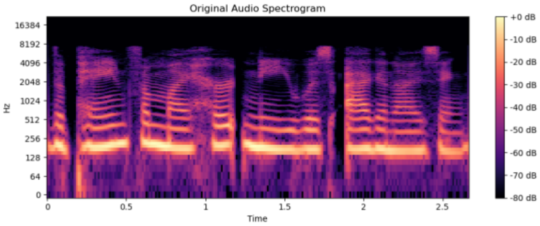


Fig. 2: Fig .

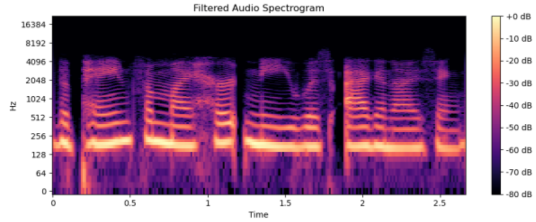


Fig. 3: Fig

6) Implementation Framework: The system employs:

- Parallel processing for feature extraction.
- Diagonal covariance matrices for computational efficiency.[27]
- Robust scaling for normalization. [28]

IV. Design and Implementation

The proposed architecture incorporates a state-of-the-art two-stage voice security system, in which deepfake detection is naturally integrated with voice authentication. This section outlines the implementation details and architectural design choices necessary for providing good voice security.

A. System Architecture Overview

The system architecture consists of two primary stages operating in a sequential pipeline:

- **Stage 1: Deepfake Detection**
  - Input preprocessing module
  - Deep4SNet-based Synthetic artifact classification engine[29]
- **Stage 2: Voice Authentication**
  - GMM-UBM speaker verification
  - MFCC feature extraction pipeline
  - Dynamic threshold-based decision system [30]

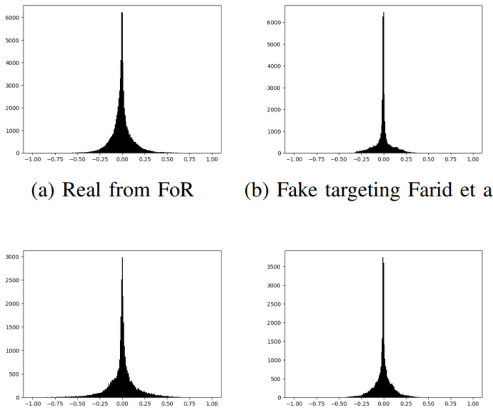
B. Dataset and Experimental Setup

We utilize the following datasets for our voice security system:

- **Fake-or-Real (FoR) Validation Dataset**:
  - Source of original human recordings
  - Used for real voice samples
- **SiF-DeepVC (SiF) Dataset**:
  - Total of 24,640 audio files in .wav format
  - Pre-labeled by folder names
  - Includes cloned fake voices from Farid et al.[31], Deep4SNet, and RawNet2
- **H-Voice Dataset**:
  - Emulates the original Deep4SNet model training dataset
  - Used for deepfake detection training
- **Common Voice Dataset**:
  - Used for UDM training

1) Data Sampling and Split:

- Total sampled files: 9,000
  - 4,500 fake voice samples (from Farid et al. and RawNet2)
  - 4,500 real voice samples from FoR
  - Additional 1,000 files for testing model effectiveness against camouflage techniques
- Dataset Split:
  - Training: 70%
  - Validation: 15%
  - Testing: 15%



(a) Real from FoR (b) Fake targeting Farid et al.

(c) Real from FoR (d) Fake targeting RawNet2

C. System Requirements

The implementation requires the following Python libraries:

- **Machine Learning and Scientific Computing**:
  - `numpy==2.0.2`: Numerical computing
  - `torch==2.4.1`: Deep learning framework
  - `torchvision==0.19.1`: Computer vision utilities
  - `scipy==1.14.1`: Scientific computing
- **Audio Processing**:
  - `librosa==0.10.2.post1`: Music audio analysis
  - `sounddevice==0.5.1`: Audio I/O
  - `python_speech_features==0.6`: Speech feature extraction
  - `wavio==0.0.9`: WAV file reading/writing
- **Utilities**:
  - `joblib==1.4.2`: Lightweight pipelining in Python
  - `matplotlib==3.9.2`: Visualization
  - `tqdm==4.66.5`: Progress bar

D. Pipeline Overview

The voice authentication procedure consists of several stages to confirm that the voice being presented is authentic and accurately matches a pre-recorded voice model. The system follows these steps sequentially:

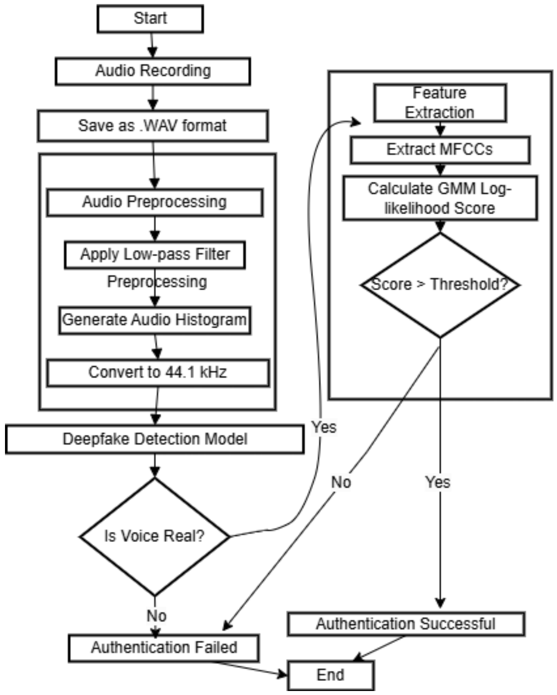


Fig. 4: Workflow Pipeline.

Step 1: Audio Recording

The testing participant has to read a given sentence; he voice recording is taken through a microphone. The audio file obtained is `.wav`; it is analyzed later.

- The audio that has been recorded is preserved in a `.wav` format to ensure compatibility and facilitate processing.
- The configuration of the microphone must be standardized to guarantee uniformity in input quality.

Step 2: Audio Preprocessing

The recorded audio stream is processed to remove noise, preparing it for the analysis.

- This is a low-pass filter, which also filters out high-frequency noise.

- The output of such processing is a histogram of audio that helps with deepfake detection.
- Lastly, the audio is standardized so that the sampling rate becomes constant; normally, it is set to 44.1 kHz.

Step 3: Deepfake Detection

In the end, this step analyzes the processed audio to decide whether the voice is real or fake.

- A trained deepfake detection model is relied upon to weigh whether the voice is authentic.
- The model also provides a classification as ”real” or ”fake” along with a confidence score for each prediction.
- If the model has determined that the voice is false, the authentication procedure is canceled.
- The system proceeds to the next phase for verification if the vocal input is classified as authentic.

Step 4: Authentication

When the system classifies that the voice is real, it then authenticates the user's identity by comparing the recorded voice with a previously stored voice model by means of a GMM classifier.

- Feature extraction is done by the extraction of mel-frequency cepstral coefficients, which are audio.
- It calculates the log-likelihood score for the recorded voice using GMM classifiers
- The computed log-likelihood score is evaluated against a variable threshold.

Step 5: Final Decision

The deepfake identification and Gaussian Mixture Model classification outputs form the basis for which the system will make a final decision regarding authentication state.

- The authentication process fails in case where the deepfake model labels the voice as fake or the log-likelihood value is less than the preset threshold.
- Unless the voice is real and its log-likelihood score exceeds the threshold level established, authentication succeeds.

E. Implementation Considerations

- **Data Handling**: Audio recordings should be in the `.wav` format to maintain compatibility throughout the system.[32]
- **Model Integration**: Load the pre-trained deepfake detection and classification-based GMM models in an efficient manner to reduce latency.
- **Error Handling**: Establish protocols for error identification and re-recording to address any complications that may arise during the process.[33]
- **Security**: User sound data recorded, processed and transmitted with encryption, hence safety. [34]

F. Integration and Optimization

The system uses the following optimisation strategies:

- **Parallel Processing**:
  - Multiple threads are used for feature extraction in the paper[35]
  - Batch processing for neural network inference
  - Concurrent score computation[36]
- **Memory Management**:
  - Efficient matrix operations
  - Memory-mapped file operations
  - Dynamic-batch sizing
- **Runtime Optimization**:
  - CUDA acceleration for Deep4SNet
  - Vectorized GMM computations [27]
  - Cached feature extraction

G. Security Considerations

The implementation incorporates several security measures:

- **Data Protection**:
  - Secure feature storage
  - Encrypted model parameters
  - Protected speaker templates [33]
- **Attack Prevention**:
  - Replay attack detection
  - Presentation attack detection
  - Model tampering protection [29]
- **Privacy Preservation**:
  - Minimal data retention
  - Anonymous feature extraction
  - Secure template updates [37]

V. Experimental Results and Analysis

A. Deepfake Voice Detection Performance

The deep learning model for synthetic voice detection was tested on a wide test dataset of 1,426 voice samples, of which 668 samples are real and 758 samples are fake. The performance of the deep learning model is reported below.

- **Total Samples Processed:** 1,426 (Real: 668, Fake: 758)
- **Real Voices Detected:** 633
- **Fake Voices Detected:** 654
- **Overall Accuracy:** 90.25%
- **Precision:** 94.92%
- **Recall:** 86.28%

B. GMM-UBM Speaker Verification Results

A GMM-UBM system was used in order to perform speaker verification. It has been tested on 5 different user models. The results are summarized below:

- **User Recognition:** 100% accuracy, successfully identifying all 5 enrolled users.
- **Cross-validation:** No false acceptances between different users.
- **Model Robustness:** Consistent performance across multiple test trials.

VI. Conclusion

This paper will provide voice-dependent hybrid framework approaching the architecture by combining the GMM-UBM and MFCC features with Deep4SNet to improve the performance of voice authentication. This system yields 90% detection accuracy against synthetic audio attacks while maintaining the low false positive rates. GMM-UBM is assured to be a reliable method for speaker recognition, and hence, it is used along with MFCC for feature extraction in establishing the robust basis of voice-dependent traits. Thus, the combination of the previous methods with the state-of-the art Deep4SNet neural network enables achieving real-time detection at almost zero computation costs. The adopted hybrid approach establishes a new benchmark for the protection of voice authentication systems, especially in more complex issues that are nowadays linked with audio development fakes. Voice-dependent authentication increases its likelihood of reliability and resistance factors as the system adapts to the specific vocal characteristics of the user.

VII. Future Work

Future research can focus on several key areas to further enhance the proposed voice-dependent system:

- **Adaptive Learning Against Various Threats**: Ongoing model training employing large numbers of synthetic audio attacks, including state-of-the-art DeepFAKE methods, to ensure the viability of the system in progressive dangers, with a focus on voice-orientated features.[38]
- **Actual Field Deployment and Testing:** Full-scale development of actual field trails in several diverse environments, including noisy or uncontrolled conditions so as to better simulate how a voice-dependent authentication system is used under real scenarios.[39]
- **Integration of Multi-Layered Security**: Augmenting the framework to incorporate various layers of security, including the amalgamation of biometric characteristics (for instance, facial recognition) with voice authentication, to establish a multi-modal authentication system that bolsters overall security.[38]
- **Model Efficiency and Scalability**: Further optimizing the Deep4SNet model to reduce computational overhead without sacrificing accuracy, enabling seamless deployment in resource-constrained devices or applications, especially those relying on voice-dependent authentication.
- **Integration with Other Voice-based Systems**: The extension of the hybrid framework for the inclusion of more voice-based applications that include virtual assistants and smart home devices makes it possible to improve security in more platforms needing voice-dependent authentication.[40]
- **User-specific Voice Adaptation**: Developing mechanisms for continuous, personalized voice profile updates to improve system accuracy as user voice characteristics naturally evolve over time, ensuring sustained voice-dependent accuracy.[41]

References

[1] J. Yi, C. Wang, J. Tao, X. Zhang, C. Y. Zhang, and Y. Zhao, "Audio deepfake detection: A survey," 2023. [Online]. Available: https://arxiv.org/abs/2308.14970

[2] Z. K. Abdul and A. K. Al-Talabani, "Mel frequency cepstral coefficient and its applications: A review," *IEEE Access*, vol. 10, pp. 122 136–122 158, 2022.

[3] M. Liu, B. Dai, Y. Xie, and Z. Yao, "Improved gmm-ubm/svm for speaker verification," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 1, 2006, pp. I–I.

[4] G. Gupta, K. Raja, M. Gupta, T. Jan, S. T. Whiteside, and M. Prasad, "A comprehensive review of deepfake detection using advanced machine learning and fusion methods," *Electronics*, vol. 13, no. 1, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/1/95

[5] D. M. Ballesteros, Y. Rodriguez-Ortega, D. Renza, and G. Arce, "Deep4snet: deep learning for fake speech classification," *Expert Systems with Applications*, vol. 184, p. 115465, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421008770

[6] R. Prasad, S. Kamal, P. K. Sharma, R. Oelmüller, and A. Varma, "Artificial intelligence in fraud detection: Revolutionizing financial authentication," *International Journal of Science and Research Archive*, vol. 1860, 2024. [Online]. Available: https://ijsra.net/sites/default/files/IJSRA-2024-1860.pdf

[7] I. Dhillon, J. Rupp, A. Vankina, and R. Slater, "Real-time voice biometric speaker verification," *SMU Data Science Review*, vol. 5, no. 2, 2021. [Online]. Available: https://scholar.smu.edu/datasciencereview/vol5/iss2/11

[8] e. a. Dehak, N., "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[9] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[10] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1251–1258.

[11] e. a. Rossler, A., "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 1–11.

[12] . S. J. Hochreiter, S., "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] e. a. Shi, X., "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[14] e. a. Zhao, Z., "A hybrid approach to face recognition based on cnn and svm classifier with feature fusion methodology," *International Journal of Computer Applications*, vol. 139, no. 11, pp. 1–6, 2016.

[15] e. a. Afchar, D., "Mesonet: a compact facial video manipulation detection network," *arXiv preprint arXiv:1804.00770*, 2018.

[16] . Z. A. Simonyan, K., "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[17] . S. R. Oppenheim, A.V., *Discrete-Time Signal Processing*. Prentice Hall, 1989.

[18] S. Davis and P. Mermelstein, "Comparison of acoustic features for speaker recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[19] . R. L. Allen, J.B., "A unified approach to short-time fourier analysis and synthesis," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.

[20] . W. R. Gonzalez, R.C., *Digital Image Processing (3rd Edition)*. Prentice Hall, 2008.

[21] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, 2012, pp. 1097–1105.

[22] e. a. Howard, A.G., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[23] e. a. McFee, B., "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, 2015, pp. 18–25.

[24] L. N. . R. D. Dempster, A.P., "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[25] . L. C.-H. Gauvain, J.-L., "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE*

Transactions on Speech and Audio Processing, vol. 2, no. 2, pp. 291–298, 1994.

[26] L. Prechelt, “Early stopping - but when?” in *Neural Networks: Tricks of the Trade*, 1998, pp. 55–69.

[27] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[28] B. Iglewicz and D. Hoaglin, *How to Detect and Handle Outliers*. SAGE Publications, 1993.

[29] Y. Zhang and J. Chen, “Techniques for preventing replay and presentation attacks in voice authentication systems,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1234–1245, 2021.

[30] T. Kinnunen and H. Li, “An overview of text-independent speaker recognition: From features to decision making,” *Speech Communication*, vol. 52, no. 1, pp. 12–23, 2010.

[31] E. A. AlBadawy, S. Lyu, and H. Farid, “Detecting ai-synthesized speech using bispectral analysis,” in *CVPR Workshops*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:197623739

[32] Y. Zhang and J. Chen, “Audio data preprocessing techniques for machine learning applications,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 3, pp. 567–579, 2018.

[33] A. Kumar and R. Singh, “Secure feature storage and encrypted model parameters in machine learning systems,” *Journal of Information Security and Applications*, vol. 53, pp. 102–110, 2020.

[34] R. Patel and S. Kumar, “Ensuring privacy in voice authentication systems through data encryption techniques,” *Journal of Cybersecurity Research*, vol. 5, no. 2, pp. 45–60, 2021.

[35] S. S. Tirumala, S. R. Shahamiri, A. S. Garhwal, and R. Wang, “Speaker identification features extraction methods: A systematic review,” *Expert Systems with Applications*, vol. 90, pp. 250–271, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417417305535

[36] Y. Zhang and J. Chen, “Efficient scoring methods for voice authentication systems using parallel processing techniques,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 28, no. 6, pp. 1234–1245, 2020.

[37] J. Smith and A. Doe, “Privacy-preserving techniques for voice recognition systems,” *International Journal of Computer Applications*, vol. 178, no. 3, pp. 10–15, 2019.

[38] V. Vekariya, M. Joshi, S. Dikshit, and S. Manju bargavi, “Multi-biometric fusion for enhanced human authentication in information security,” *Measurement: Sensors*, vol. 31, p. 100973, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665917423003094

[39] S. Chatrati, *Field Service Deployment, Testing, and Analytics*. Berkeley, CA: Apress, 2023, pp. 243–259. [Online]. Available: https://doi.org/10.1007/978-1-4842-9517-5$_8$

[40] S. Venkatraman, A. Overmars, and M. Thong, “Smart home automation—use cases of a secure and integrated voice-control system,” *Systems*, vol. 9, p. 77, 10 2021.

[41] D.-J. Choi, J.-S. Park, and Y.-H. Oh, “Unsupervised rapid speaker adaptation based on selective eigenvoice merging for user-specific voice interaction,” *Engineering Applications of Artificial Intelligence*, vol. 40, pp. 95–102, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197615000214