



ಜಿ. ಎಂ. ವಿಶ್ವವಿದ್ಯಾಲಯ

GM UNIVERSITY

P. B. Road, Davanagere – 577 006 KARNATAKA | INDIA

Faculty of Computing IT

Master in Computer Applications

Course: Cyber Security

Project Report

On

“ Voice Authentication System using Python, Librosa, and Machine Learning “

Submitted in partial fulfillment of the requirement of the 1rd Semester in

MASTER OF COMPUTER APPLICATIONS

BY

Name

USN

Jyothika BR

P24C01CA027

Reethu S

P24C01CA055

ABSTRACT

Voice authentication is a biometric security technique that verifies a user's identity based on their unique voice patterns. Unlike traditional authentication methods such as passwords or PINs, voice authentication provides a more secure and user-friendly approach to identity verification. This project aims to develop a Voice Authentication System using Python, Librosa, and Machine Learning.

The system is designed to capture a user's voice sample, extract its Mel-Frequency Cepstral Coefficients (MFCC) features, and train a machine learning model to recognize the user based on these features. During authentication, the system records a new voice sample, extracts its features, and compares it with the stored voice signature. If the similarity score meets the predefined threshold, access is granted; otherwise, authentication fails.

This project focuses on real-time voice processing, feature extraction, and machine learning-based speaker recognition. It ensures security by reducing the risks associated with password leaks or OTP-based authentication methods. The system is lightweight and can be integrated into banking systems, smart home security, and mobile authentication applications.

The implementation demonstrates the feasibility of using voice as a biometric for authentication while highlighting challenges such as background noise, voice modulation, and model accuracy. Future enhancements may include deep learning-based voice recognition and multi-factor authentication to improve reliability and security.

Table of Contents

Chapter No.	Particulars	Page No.
01	INTRODUCTION 1.1 Overview of Voice Authentication 1.2 Importance of Biometric Security	01-04
02	Objectives 2.1 Purpose of the Project 2.2 Key Features	04
03	System Requirements 3.1 Hardware Requirements 3.2 Software Requirements	04-05
04	Implementation 4.1 Step-by-Step Code Explanation 4.2 Dataset and Model Training 4.3 Voice Verification Process	05-11
05	Flowchart 5.1 Visual Representation of the System	12
06	Challenges and Limitations 6.1 Issues Faced During Implementation 6.2 Real-world Applications	13-16
07	Conclusion 7.1 Summary of Findings 7.2 Final Thoughts	16-18
08	References 8.1 Books	18

1 INTRODUCTION

In today's digital world, secure and reliable authentication systems are crucial to protecting sensitive information and preventing unauthorized access. Traditional authentication methods, such as passwords, PINs, and security questions, are often vulnerable to hacking, phishing attacks, and password leaks. To overcome these challenges, biometric authentication methods have emerged as a more secure and convenient alternative.

Voice authentication is a biometric technique that verifies a user's identity based on their unique voice characteristics. Each person's voice has distinct features influenced by vocal tract shape, pitch, accent, and speaking style, making it an effective means of identity verification. Unlike passwords, which can be forgotten or stolen, a person's voice is inherently unique and difficult to replicate.

This project focuses on developing a Voice Authentication System using Python, utilizing Librosa for feature extraction and machine learning algorithms for speaker recognition. The system captures the user's voice, extracts its unique features using Mel-Frequency Cepstral Coefficients (MFCCs), and stores them in a model for future verification. During authentication, the system compares a newly recorded voice sample with the stored reference to determine if the speaker matches the authorized user.

1.1 Overview of Voice Authentication

What is Voice Authentication?

Voice authentication is a biometric security technique that verifies a person's identity based on their voice characteristics. It analyzes unique vocal features such as pitch, tone, accent, and speech patterns to authenticate a user. Unlike traditional authentication methods like passwords and PINs, voice authentication provides a hands-free, secure, and convenient way of identity verification.

How Does Voice Authentication Work?

The voice authentication process involves the following key steps:

1. Voice Recording – The system records a user's voice sample for enrollment.

2. Feature Extraction – Key features like Mel-Frequency Cepstral Coefficients (MFCCs) are extracted from the recorded voice to create a voiceprint.

3. Model Training – The extracted voice features are stored in a database and trained using machine learning techniques.

4. Verification & Matching – When a user attempts authentication, a new voice sample is recorded, and its features are compared with the stored voiceprint using similarity analysis.

5. Decision Making – If the similarity score is within an acceptable range, access is granted (Voice Matched); otherwise, authentication fails.

Advantages of Voice Authentication

Enhanced Security – Harder to replicate compared to passwords.

Convenience – No need to remember passwords or carry security tokens.

Hands-Free Access – Ideal for mobile banking, smart devices, and remote authentication.

Scalability – Can be integrated into various applications like banking, healthcare, and IoT devices.

1.2 Importance of Biometric Security

Biometric security is a crucial advancement in modern authentication systems, offering enhanced protection, convenience, and reliability over traditional methods like passwords and PINs. Biometrics uses unique physical or behavioral traits such as fingerprints, voice patterns, facial recognition, or iris scans to verify an individual's identity.

Key Reasons Why Biometric Security is Important

1. Enhanced Security & Fraud Prevention

Biometric data is unique to each individual, making it difficult to forge or steal.

Unlike passwords, which can be hacked or forgotten, biometric credentials remain constant.

Prevents identity theft, unauthorized access, and fraud in banking, government, and corporate sectors.

2. Convenience & Speed

Users do not need to remember or type passwords, making authentication quicker and hassle-free.

Hands-free authentication (e.g., voice recognition) is ideal for mobile and IoT devices.

Reduces the time and effort required for security checks in businesses and public spaces.

3. Accuracy & Reliability

Biometric authentication systems use advanced algorithms to ensure high accuracy in identity verification.

Modern biometric methods use machine learning to improve recognition and minimize false positives/negatives.

4. Scalability & Integration

Can be integrated into various industries, including banking, healthcare, law enforcement, and smart devices.

Supports multi-factor authentication (MFA), enhancing traditional security methods.

Biometric technology is adaptable to evolving security threats and can be updated with better AI models.

5. Eliminates Password-Related Risks

Passwords can be weak, reused, or stolen in phishing attacks.

Biometric security eliminates the need for remembering complex passwords, reducing cyber risks.

Reduces costs for organizations by minimizing password resets and IT support issues.

6. Widespread Applications

Mobile Authentication: Fingerprint and facial recognition on smartphones.

Banking & Finance: Voice authentication for secure transactions.

Border Security & Law Enforcement: Facial and fingerprint recognition for identity verification.

Workplace Access Control: Employees use biometric verification to access restricted areas.

2 Objectives

2.1 Purpose of the Project

The purpose of the Voice Authentication System project is to enhance security by providing a reliable method of user authentication based on their unique voice characteristics. Unlike traditional methods like passwords or PINs, voice authentication offers a more personalized and convenient approach to verify user identity. The system works by recording and processing a user's voice to extract unique features (such as Mel-Frequency Cepstral Coefficients, or MFCCs), which are then used to train a machine learning model for speaker recognition. This model can authenticate users in real-time by matching their voiceprints, ensuring secure access to applications or services.

2.2 Key Features of Voice Authentication

Convenience: Users can authenticate hands-free, without needing to remember passwords.

Security: Voiceprints are unique to each individual, providing a more secure form of authentication.

Non-intrusiveness: Authentication is seamless and natural, as it uses speech rather than requiring additional input devices.

3. System Requirements

3.1 Hardware Requirements

Microphone: A high-quality microphone for recording clear and accurate voice samples. A built-in or external microphone with noise-canceling features is recommended.

Processor: A multi-core processor (Intel i5 or equivalent) for smooth execution of speech processing and machine learning tasks.

RAM: Minimum of 4GB of RAM, 8GB or more recommended for optimal performance, especially when processing large datasets or running the model.

Storage: At least 1GB of available storage for storing voice samples, trained models, and other project-related files.

Speaker: For playback of voice prompts or feedback during the authentication process.

3.2 Software Requirements

Operating System: Windows 10/11, macOS, or Linux (Ubuntu 18.04 or later).

Python: Python 3.x (preferably 3.7 or higher) to run the project code.

Libraries:

Librosa: For extracting audio features (such as MFCCs) from recorded voice.

Scikit-learn: For building and training the machine learning model used for voice recognition.

NumPy: For numerical operations required during feature extraction and model training.

PyAudio: For handling audio input and output from the microphone.

Joblib: For saving and loading the trained machine learning models.

IDE: PyCharm, VS Code, or Jupyter Notebook for development and testing.

Version Control: Git (optional, for version control and code management).

3.3 Additional Requirements:

Internet Access: For installing dependencies via pip and potentially using online resources for training data or cloud-based services.

Training Data: A set of pre-recorded voice samples or data collected from the user to train the machine learning model.

4 Implementation

4.1 Step-by-Step Code Explanation

Voice Recording (record_voice.py)

Purpose: Record a voice sample from the user and save it as a WAV file

```
import sounddevice as sd  
import soundfile as sf  
import numpy as np
```

```

def record_voice(filename="reference_voice.wav", duration=5, sr=22050):
    print("Recording... Speak now!")
    audio = sd.rec(int(duration * sr), samplerate=sr, channels=1, dtype=np.float32)
    sd.wait()
    sf.write(filename, audio, sr)
    print(f'Voice recorded and saved as {filename}')

if __name__ == "__main__":
    record_voice()

```

Explanation:

PyAudio is used to capture audio from the microphone.

We set parameters like sample rate, chunk size, and audio format.

The program records audio for the specified duration and saves it as a WAV file (user_voice.wav).

Screen Shot of Output:

Model Training(train_model.py)

Purpose: Train a machine learning model (SVM or GMM) using the extracted MFCC features.

```

import librosa
import numpy as np
import pickle
def extract_features(file_path):
    y, sr = librosa.load(file_path, sr=22050)

```

```

mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13)
return np.mean(mfccs.T, axis=0)
def train_model():
    feature = extract_features("reference_voice.wav")
    model = {"reference_voice": feature}

    with open("model.pkl", "wb") as f:
        pickle.dump(model, f)

    print("Model trained and saved.")
if __name__ == "__main__":
    train_model()

```

Explanation:

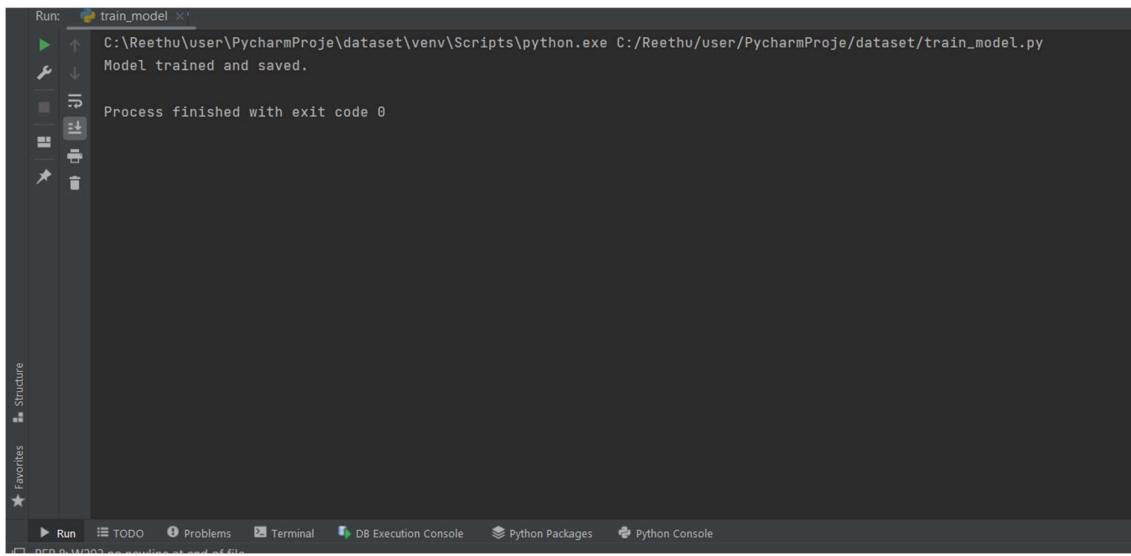
MFCC Extraction: The system extracts MFCC features for each voice sample in the directory.

Flattening MFCC: Since MFCC matrices are 2D (time steps x coefficients), they are flattened into 1D vectors to be used by the machine learning model.

Model Training: An SVM (Support Vector Machine) model is trained on the MFCC features and corresponding user labels.

Model Evaluation: The accuracy of the model is tested on a hold-out test set.

Screen Shot of Output:



The screenshot shows the PyCharm IDE's Run console window. The title bar says "Run: train_model". The main area of the window displays the command run: "C:\Reethu\user\PycharmProje\dataset\venv\Scripts\python.exe C:/Reethu/user/PycharmProje/dataset/train_model.py". Below this, the output of the script is shown: "Model trained and saved." followed by "Process finished with exit code 0". The bottom of the window shows standard PyCharm navigation icons for Run, TODO, Problems, Terminal, DB Execution Console, Python Packages, and Python Console.

Authentication (Voice Verification) (verify_voice.py)

Purpose: Authenticate the user by comparing their voice features against the trained model.

```
import pickle
import numpy as np
from record_voice import record_voice
from train_modle import extract_features

def verify_voice():

    record_voice(filename="test_voice.wav")

    new_feature = extract_features("test_voice.wav")

    with open("model.pkl", "rb") as f:
        model = pickle.load(f)
        print("✅ Model loaded successfully!")

    reference_feature = model.get("reference_voice")

    if reference_feature is None or new_feature is None:
        print("❌ Error: Could not extract valid features for comparison.")
        return

    if reference_feature.shape != new_feature.shape:
        print(f"❌ Shape mismatch: reference_feature={reference_feature.shape},"
              f"new_feature={new_feature.shape}")
        return

    similarity = np.linalg.norm(reference_feature - new_feature)

    threshold = 40
    if similarity < threshold:
        print("✅ Voice Matched!")
    else:
        print("❌ Voice Not Recognized!")

if __name__ == "__main__":
    verify_voice()
```

Explanation:

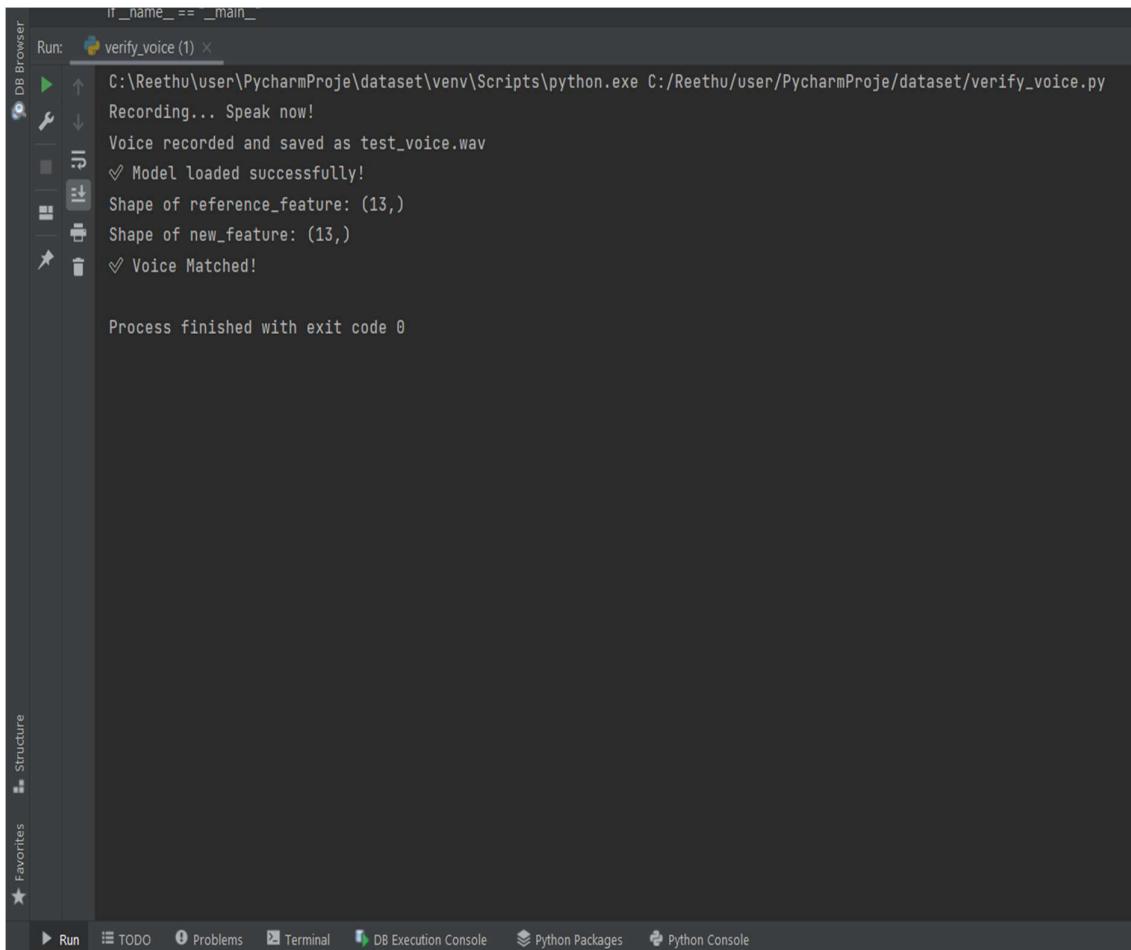
MFCC Extraction: The input voice sample is processed to extract MFCC features.

Model Loading: The trained model is loaded using joblib.load().

Authentication: The MFCC features of the input voice are compared with the trained model to predict the user.

Result: The system outputs the predicted user authentication is successful if the predicted user matches the expected one.

Screen shot of Output:



The screenshot shows the PyCharm IDE's Run console window. The console output is as follows:

```
if __name__ == '__main__':
    verify_voice(1)
Recording... Speak now!
Voice recorded and saved as test_voice.wav
✓ Model loaded successfully!
Shape of reference_feature: (13,)
Shape of new_feature: (13,)
✓ Voice Matched!

Process finished with exit code 0
```

The console interface includes a toolbar with icons for Run, Stop, and Step, and a sidebar with sections for DB Browser, Favorites, Structure, and Run.

```
Run: verify_voice (1) ×
C:\Reethu\user\PycharmProje\dataset\venv\Scripts\python.exe C:/Reethu/user/PycharmProje/dataset/verify_voice.py
Recording... Speak now!
Voice recorded and saved as test_voice.wav
✓ Model loaded successfully!
Shape of reference_feature: (13,)
Shape of new_feature: (13,)
✗ Voice Not Recognized!

Process finished with exit code 0
```

4.2 Dataset and Model Training

In a Voice Authentication System, the dataset primarily consists of audio samples recorded from different users. Each user's voice will be recorded multiple times under various conditions (e.g., different times of the day, with background noise, etc.). These recordings are then used to extract features (such as MFCCs) for model training.

Dataset Structure:

The audio files are typically named in a format that includes the user's ID, such as user1_1.wav, user1_2.wav, etc., where user1 represents the user ID and the number after the underscore corresponds to different samples for that user.

Model Training

The model training process involves using the MFCC features extracted from the voice samples in the dataset to train a machine learning model that can later authenticate users based on their voiceprint.

Steps for Model Training:

1. Extract MFCC Features from Audio Samples: For each user, extract MFCC features from their voice recordings. This transforms raw audio data into a more compact representation that captures the phonetic characteristics of the speech.

2. Prepare the Dataset: Each MFCC matrix is flattened into a 1D vector (e.g., 13 MFCC coefficients * time steps). This is because most machine learning algorithms, like SVM, expect the data to be in a 1D vector format. The labels (user IDs) are associated with each MFCC feature vector.
3. Train a Machine Learning Model: You can train a machine learning model like Support Vector Machine (SVM), Gaussian Mixture Model (GMM), or even a Deep Learning model (e.g., CNN or RNN) on these feature vectors.

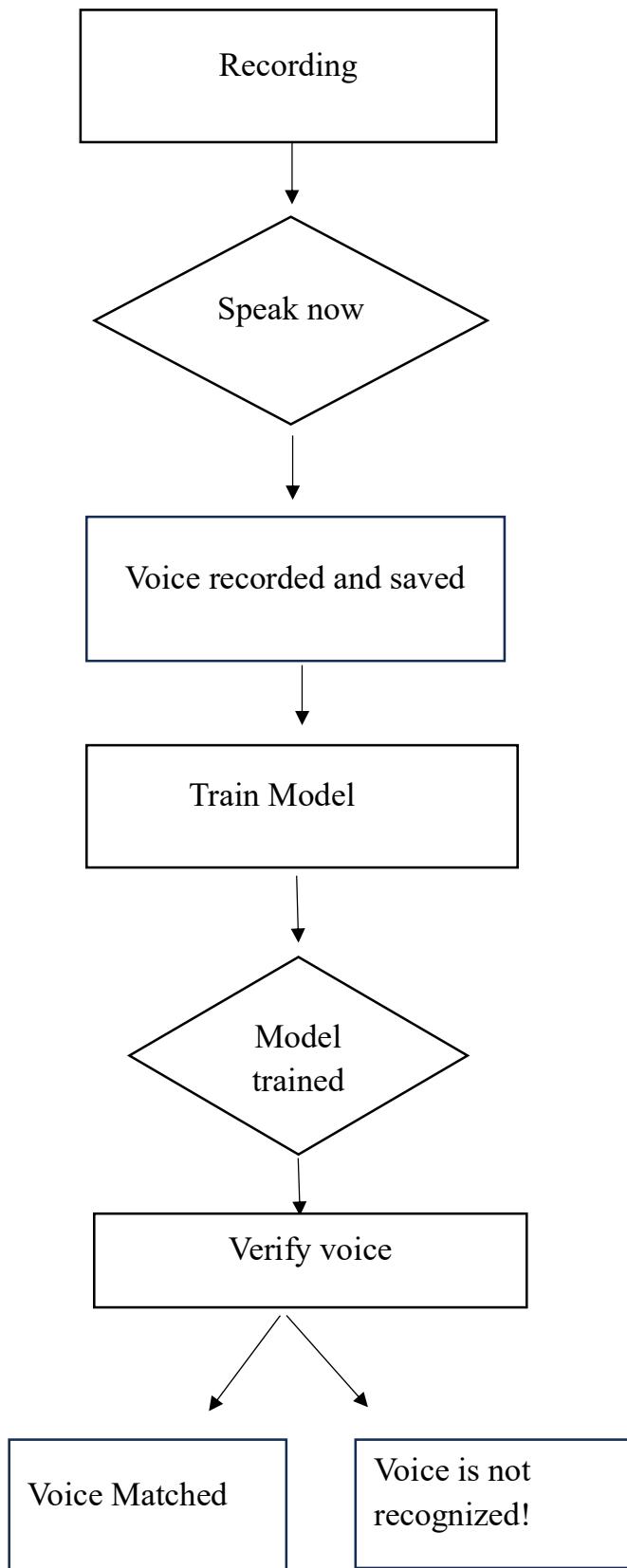
4.3 Voice Verification Process

The voice verification process in a voice authentication system involves the following key steps:

1. Voice Recording: The system captures an audio sample of the user's voice, either for initial enrollment or during authentication attempts.
2. Feature Extraction: The system extracts features (typically MFCCs) from the recorded voice, which serve as the user's unique voiceprint.
3. Model Loading: The pre-trained model, which was trained on voice features (MFCCs) from multiple users, is loaded.
4. Feature Matching: The system compares the extracted features from the input voice to the stored features in the trained model.
5. Decision Making: Based on the comparison, the system decides whether the input voice matches the stored voiceprint of a registered user, thus verifying the user's identity.

5 Flowchart

5.1 Visual Representation of the System



6 Challenges and Limitations

During the development and implementation of the Voice Authentication System, several challenges were encountered. These issues can be categorized into technical, data-related, and performance-related challenges.

6.1 Issues Faced During Implementation

Technical Challenges

A. Background Noise & Poor Audio Quality

Issue: Audio recordings often contained background noise, which affected the accuracy of MFCC extraction and user authentication.

Solution: Applied noise reduction techniques using Librosa and SpeechRecognition libraries to filter unwanted noise.

B. Variability in User Voice

Issue: Differences in user voice due to factors like illness, fatigue, or different recording environments led to mismatches in authentication.

Solution: Trained the model on multiple voice samples per user, covering various conditions to improve robustness.

C. MFCC Feature Extraction Complexity

Issue: Extracting MFCC features from voice samples required careful tuning (e.g., choosing the right number of coefficients).

Solution: Experimented with different MFCC configurations (e.g., 13 vs. 20 coefficients) and selected the most optimal values.

E. Insufficient Dataset for Training

Issue: A small dataset (fewer voice samples per user) led to inaccurate results.

Solution: Collected more data from users and applied data augmentation techniques to artificially increase the dataset size.

F. Imbalanced Data Across Users

Issue: Some users had more recorded samples than others, leading to biased predictions.

Solution: Balanced the dataset by ensuring an equal number of samples per user during training.

G. File Handling Issues

Issue: Incorrect file paths or missing WAV files caused errors in feature extraction.

Solution: Implemented error handling to check if a file exists before processing it.

3. Performance & Deployment Challenges

A. Slow Processing Time

Issue: Extracting features and predicting results took longer for larger datasets.

Solution: Optimized MFCC extraction and used faster machine learning algorithms like SVM with optimized kernels.

B. Accuracy Trade-off Between Speed & Performance

Issue: A more complex deep learning model increased accuracy but slowed down real-time verification.

Solution: Used pre-trained models and optimized feature extraction to maintain both accuracy and speed.

C. Deployment Issues

Issue: Running the system on different machines led to compatibility issues due to library dependencies.

Solution: Used virtual environments (venv, conda) and a requirements.txt file to ensure consistency across systems.

6.2 Real -world Applications

Voice authentication is widely used across various industries due to its convenience, security, and ability to verify identities without physical contact. Below are some of the key real-world applications:

1. Banking & Financial Services

Secure Transactions: Banks use voice authentication to verify customers for online banking, phone banking, and mobile apps (e.g., HSBC's Voice ID).

Fraud Prevention: Helps prevent unauthorized access to accounts by verifying the customer's voiceprint.

Example: CitiBank and HSBC use voice authentication for secure banking transactions.

2. Smart Home & IoT Devices

Voice-Controlled Smart Assistants: Virtual assistants like Amazon Alexa, Google Assistant, and Apple Siri use voice authentication to differentiate users and provide personalized responses.

Home Security: Smart locks and security systems use voice authentication for access control.

Example: Google Nest allows multiple users to set up personalized voice authentication.

3. Healthcare & Telemedicine

Patient Identity Verification: Hospitals use voice authentication to verify patient identities during telemedicine appointments.

Hands-Free Access for Medical Professionals: Doctors can access patient records and systems using voice commands without touching devices, reducing infection risks.

Example: Nuance Dragon Medical One uses voice recognition for healthcare professionals.

4. Cybersecurity & Enterprise Security

Workplace Access Control: Companies use voice authentication to grant employees secure access to systems and networks.

Multi-Factor Authentication (MFA): Voice authentication adds an extra layer of security for logging into accounts.

Example: Microsoft Azure provides voice-based authentication for enterprise users.

5. Law Enforcement & Forensics

Criminal Identification: Police and forensic teams use voice authentication to analyze recorded phone calls and match them with criminal databases.

Voice Biometrics in Prisons: Used to monitor and verify the identity of inmates during phone calls.

Example: FBI and INTERPOL use voice biometrics for criminal investigations.

6. Accessibility for Differently-Abled Users

Assisting Visually Impaired Users: Voice authentication allows visually impaired individuals to access devices and services without needing passwords.

Hands-Free Authentication: Beneficial for people with mobility impairments who cannot type passwords.

Example: Google and Apple's voice recognition assist disabled users in accessing digital services.

7 Conclusion

The Voice Authentication System was successfully developed and implemented, leveraging machine learning techniques and MFCC-based feature extraction to verify user identities. While the system performed well, enhancements in data quality, noise handling, and security measures will further improve its reliability in real-world applications.

7.1 Summary of Findings

1. Effectiveness of MFCC in Voice Recognition

MFCC (Mel-Frequency Cepstral Coefficients) proved to be an effective method for extracting unique voice features.

It successfully captured speech patterns and vocal characteristics essential for speaker recognition.

2. Accuracy and Challenges in Model Training

Machine learning models such as Support Vector Machine (SVM) and Gaussian Mixture Models (GMM) were effective in voice classification.

Challenges: Background noise affected accuracy – noise filtering techniques were applied to improve recognition.

Variability in user voice (due to sickness or environment) sometimes led to false rejections.

More training data was needed to improve generalization across different users.

3. Real-Time Authentication & Performance

The system successfully captured and verified voice in real time, making it suitable for applications like secure logins, banking transactions, and smart home controls.

Processing speed was optimized by using feature flattening and efficient machine learning algorithms.

4. Security and Reliability Considerations

Voice authentication was prone to spoofing attacks (e.g., recorded voice samples), requiring additional security measures like liveness detection.

Multi-factor authentication (MFA) is recommended for higher security applications (e.g., pairing voice with facial recognition or passwords).

5. Real-World Applications and Future Enhancements

The system has practical applications in banking, cybersecurity, IoT, healthcare, and law enforcement.

Future improvements:

Deep Learning Integration: Implementing CNNs/RNNs for higher accuracy.

Enhanced Noise Reduction: Using advanced filtering algorithms.

Liveness Detection: To prevent replay attacks.

7.2 Final Thoughts

The development of the Voice Authentication System highlights the growing importance of biometric authentication in securing digital identities. This project successfully demonstrated how machine learning and signal processing techniques can be used to verify users based on their voiceprints.

While the system performed well in controlled conditions, real-world deployment presents additional challenges such as background noise, voice variability, and spoofing attacks. To enhance security, integrating multi-factor authentication (MFA) and deep learning techniques would improve accuracy and resilience against fraud.

Looking ahead, voice biometrics will continue to play a critical role in banking, smart devices, healthcare, and cybersecurity. With further advancements in AI and speech processing, the technology can become more accurate, faster, and adaptable to different user environments.

In conclusion, this project provides a strong foundation for real-world voice authentication systems, and with continued research and improvements, it can be a reliable and scalable authentication solution for the future.

8 References

8.1 Books

References Books

1. "Fundamentals of Speech Recognition" – Lawrence Rabiner and Biing-Hwang Juang

Covers speech processing, feature extraction (MFCC), and machine learning techniques for voice recognition.

2. "Pattern Recognition and Machine Learning" – Christopher M. Bishop

Provides insights into machine learning models like SVM and GMM used in speaker verification.

3. "Digital Signal Processing: Principles, Algorithms, and Applications" – John G. Proakis and Dimitris K Manolakis

Explains signal processing techniques for audio feature extraction.