Indian Overseas Bank Voice Authentication System

Team Name: True Sight

Team ID: 09

Hackathon: IOB CYBERNOVA 2025

Mentor: Mrs. Amsavalli S, Assistant Professor

Members:

- Syed Thufel Syed Wahid (230171601189)
- Vishwanathan M (230171601196)
- Sabilah S (230171601159)
- Nizamutheen (230171601153)

Overview

This project implements a robust, **voice authentication** system designed to deliver secure, accessible, and efficient user verification without the need for passwords or physical tokens. Users simply speak a dynamically generated verification code into any microphone-equipped device, and the system processes the input in real-time using:

- CNN-Based Speaker Recognition: A Convolutional Neural Network model trained on MFCC features to identify speaker-specific voice patterns.
- MFCC Extraction & Preprocessing: Automatic trimming, normalization, and bandpass filtering ensure consistent feature quality across diverse audio sources.
- Ensemble Verification: Combines CNN predictions with average cosine similarity scores and Dynamic Time Warping (DTW) distances for resilient decision-making, even in noisy environments.

Key Benefits:

- Passwordless & User-Friendly: Eliminates the need for memorized credentials or typing, reducing friction and risk.

- Scalable & Lightweight: Operates on minimal infrastructure, making it suitable for web, mobile, and IoT deployments.
- Adaptive: Retrains the CNN model automatically upon each new registration, continuously improving accuracy as the user database grows.
- Secure: Dynamic verification codes combined with multi-metric validation (an ensemble of CNN prediction confidence, average cosine similarity, and DTW distance) mitigate replay attacks and spoofing.
- Extensible: Easily integrates with additional modalities (e.g., face recognition) and third-party systems (e.g., banking APIs).

Technologies Used

- Python 3.11+: Core programming language
- Flask: RESTful API framework
- MongoDB (PyMongo): Database for storing voice feature embeddings
- TensorFlow/Keras: CNN model development and inference
- Librosa: Audio processing and MFCC extraction
- pydub & FFmpeg: WebM-to-WAV conversion
- fastdtw: Efficient DTW calculations
- scikit-learn: Cosine similarity and data utilities
- Flask-Cors: Cross-origin support for frontend

How It Works

---

1. Request Verification Code
    - User submits user_id to /api/verification-code. A six-digit code is generated and stored in session.
2. Speak & Verify Code
    - Frontend records user speaking the code into a WebM file.

    - Backend converts WebM to WAV, extracts and normalizes MFCC features.
3. Registration (`api/register`)
    - Captures 10 MFCC samples per new user; stores in MongoDB which makes our model to train itself in a good manner to provide accurate authentication.
    - Retrains CNN on all users' data for improved accuracy.
4. Authentication (`api/verify`)
    - Extracts MFCC from incoming audio clip.
    - Loads/retrains CNN model, predicts "user_id" and confidence score.
    - Computes average cosine similarity and average DTW distance against stored samples.
    - Applies weighted ensemble: CNN (50%), Cosine (25%), DTW (25%).
    - Returns success if overall score ≥ 0.75.
5. Prerequisites

- MongoDB server running locally (mongodb://localhost:27017/)
- FFmpeg installed and configured in .env
- Python packages (see requirements)

Installation

---

1. Clone the repository:
    - git clone https://github.com/KILLERxNARUTO/truesight.git
    - cd truesight
2. Create and activate a virtual environment:
    - python -m venv venv
    - source venv/bin/activate # Mac/Linux
    - venv\Scripts\activate # Windows
3. Install dependencies:
    - pip install -r requirements.txt
    - Copy .env and set paths:
4. FFMPEG_PATH=/usr/bin/ffmpeg

Running the Project

---

1. Start MongoDB:
    - mongod --dbpath /path/to/db
2. Launch Flask API:
    - python app.py
3. Expose via Ngrok:
    - ngrok http 5000
4. Open frontend at Ngrok URL.

Run Ngrok to access frontend

---

1. Install Ngrok:
    - Download Ngrok from https://ngrok.com/download and extract the executable for your platform.
    - Add the Ngrok folder to your system PATH to enable command-line usage.
2. Start Ngrok:
    - Open CMD (Windows) or Terminal (Mac/Linux) and navigate to your project directory.
    - Run:
        - ngrok http 5000
        - Ngrok will display Forwarding URLs (HTTP and HTTPS) that tunnel to your local server.
3. Update Flask Code to Use Ngrok URL:
    - In app.py, update the CORS origins list to include your Ngrok HTTPS URL:

      CORS(app, supports_credentials=True, origins=[

      "http://localhost:5000",

      "https://<your-ngrok-url>"])

    - Replace any hardcoded API endpoint or frontend URLs in your code with https://<your-ngrok-url> for public access.
4. Update Frontend to Use Ngrok URL:
    - In your HTML frontend (in login.html,register.html and dashboard.html), search for any hardcoded backend URLs and replace them with your generated Ngrok HTTPS URL

      (e.g., https://<your-ngrok-url>).
5. Test the Ngrok Connection:
    - Open https://<your-ngrok-url> in a browser or API client to verify that your Flask app is reachable from the internet.

Troubleshooting

- WebM conversion errors: Check FFmpeg paths in .env.
- Model load failures: Ensure MODEL_PATH points to a valid .keras file.
- Authentication accuracy: Noise may degrade MFCC quality; encourage quiet input due to minimal data access.
- Database errors: Verify MongoDB is running and URI is correct.

Notes

- Sessions store temporary verification codes; clear upon logout.
- CNN retraining occurs after each new registration to adapt to new users.
- MFCC features are zero-padded/resized to uniform shape (100×40).
- Ensure that you add the pre-trained model in a right location to use that for authentication.