



## Securing Voice and Multimodal AI Agents Against Deepfakes and Prompt Injection.

### Challenge Overview

Voice-enabled and multimodal AI agents are increasingly deployed in real-world systems such as virtual assistants, customer support bots, authentication systems, and automated decision-making platforms. These agents rely on audio, visual, and textual inputs to understand user intent and trigger actions.

Recent advances in generative AI have enabled highly realistic audio deepfakes, voice cloning, and adversarial multimodal inputs. Attackers can manipulate AI agents using synthetic voices, hidden audio commands, or visual prompt injections embedded in images and videos. Traditional security mechanisms are insufficient to defend against these emerging threats.

This challenge focuses on building a robust security framework that detects and mitigates malicious audio and visual inputs targeting multimodal AI agents.

### Objectives

#### Audio Deepfake and Voice Cloning Detection

- Detect AI-generated or spoofed audio inputs.
- Identify voice cloning attacks and replay attacks.
- Analyze acoustic and spectral features for authenticity verification.

#### Multimodal Prompt Injection Detection

- Detect hidden or adversarial instructions embedded in images or videos.
- Identify mismatches between audio, visual, and textual modalities.
- Prevent unauthorized actions triggered by manipulated inputs.

#### Risk Scoring and Mitigation

- Assign a confidence or risk score to each input.
- Block, flag, or downgrade suspicious inputs before agent execution.
- Provide explanations for detected threats.

## Evaluation Criteria

### Detection Accuracy

- Precision, recall, F1-score, and AUC-ROC for deepfake detection.
- Effectiveness against unseen or novel attack samples.

### False Positive and False Negative Rates

- Incorrect classification of genuine human inputs.
- Missed detection of malicious audio or visual attacks.

### System Performance

- Latency introduced per input.
- Suitability for real-time or near-real-time applications.

### Explainability

- Clarity of detection reports.
- Interpretability of model decisions and risk scores.

## Implementation Guidelines and Suggestions

### General Implementation Rules

- Core detection and mitigation mechanisms must be implemented from scratch.
- Copying full solutions from Kaggle, GitHub, or similar platforms is strictly prohibited.
- Participants must clearly document models, features, and system architecture.

### Dataset Usage

- Teams may use publicly available datasets, such as:
  - ASVspoof 2019 / 2021 (audio spoofing detection)
  - FakeAVCeleb (audio-visual deepfake dataset)
  - LibriSpeech (genuine speech samples)
- Synthetic adversarial samples may be generated for evaluation purposes.

### Attack Scenarios

Participants should consider defending against:

- Voice cloning attacks impersonating trusted individuals.
- Replay attacks using recorded audio.
- Hidden audio commands embedded in background noise.
- Visual prompt injection via text embedded in images or videos.
- Cross-modal attacks combining benign visuals with malicious audio.

## Advanced Models and Techniques

### Audio Analysis Techniques

- Spectrogram-based CNN models for deepfake detection.
- Speaker verification and embedding comparison.
- Frequency-domain and phase-based feature analysis.

### Vision and Multimodal Analysis

- Vision-language models (CLIP, BLIP) for image-text alignment.
- Detection of hidden or low-contrast text in images.
- Cross-modal consistency checks.

### Ensemble and Hybrid Approaches

- Combine audio, visual, and textual detectors.
- Fuse rule-based and machine learning approaches.

### Scalability Considerations

- Optimize feature extraction pipelines for real-time processing.
- Support batch and streaming input modes.

## Deliverables

### Working Prototype

- A functional multimodal security system.
- Demonstration of detection against multiple attack types.
- Structured risk and detection reports.

### Source Code

- Modular, well-commented implementation.
- Repository link (GitHub/GitLab) with setup instructions.

### Documentation

- Architecture and threat model description.
- Explanation of feature extraction and models used.
- Evaluation metrics and results.

### Demo Video

- 5–10 minute video showcasing deepfake attacks and detection.
- Explanation of system decisions and outputs.

## Submission Format

- **Platform:** Submissions must be made via official college email IDs.
- **File Structure:**
  - Source code folder
  - Documentation files (`README.md`, `TechnicalDocumentation.pdf`)
  - Dataset details or preprocessing scripts (if applicable)
  - Demo video link (YouTube, Vimeo, or Google Drive)

