

# **HAVDEF: Hindi Audio Deepfake Defence**

## **Capstone Project Report**

### **MID-SEMESTER EVALUATION**

#### **Submitted by:**

(102203191) Shivane Kapoor  
(102203194) Kaustubh Singh  
(102203205) Japneet Singh  
(102203499) Arpit Jain  
(102253002) Diwakar Narayan

**BE Third Year, CoE/CoSE**

**CPG No: 207**

Under the Mentorship of

Dr. Seema Bawa (Professor)

Dr. Sandeep Verma (Assistant Professor)

Dr. Sachin Kansal (Associate Professor)



**Computer Science and Engineering Department**  
**Thapar Institute of Engineering and Technology, Patiala**  
**August 2025**

## ABSTRACT

---

This project presents **HAVDEF (Hindi Audio Deepfake Defence)**, a proposed real-time deepfake voice detection system tailored for **Hinglish code-mixed speech** in high-risk scenarios such as financial scams and impersonation fraud. The system is designed to address the growing misuse of AI-generated voices by enabling timely detection of synthetic speech directly on mobile devices, without reliance on cloud processing.

Our approach begins with the development of a **custom Hinglish dataset** that includes real and synthetic audio samples, annotated at the **phoneme and prosodic level** to capture code-switched speech characteristics. Data preprocessing techniques such as **denoising, codec-aware filtering, and augmentation** (e.g., pitch shifts, noise injection) are planned to improve robustness against varied environments and messaging platform compression.

The detection engine will integrate **spectral (Mel-spectrogram, MFCC), temporal, and prosodic features**, analysed using a **hybrid Transformer CNN model** optimised for **on-device inference** through pruning and quantisation. To strengthen resilience, **adversarial training** methods will be explored to defend against emerging deepfake generation techniques.


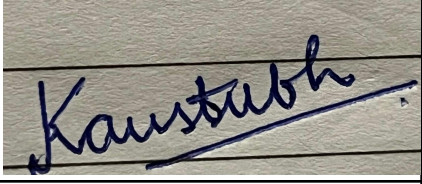
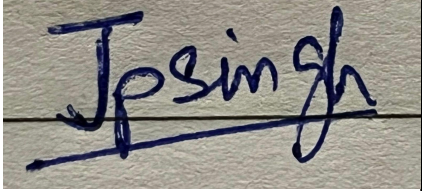
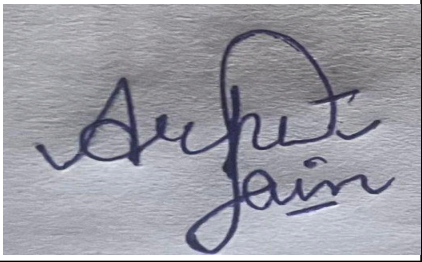
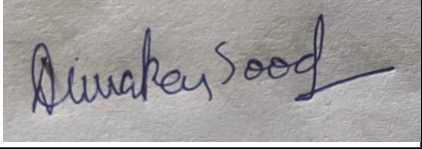
Once implemented, HAVDEF is expected to provide a **privacy-preserving, mobile-ready defence system** that alerts users in real time during suspicious calls. Beyond Hinglish, the framework has the potential to extend to **multilingual voice deepfake detection**, supporting applications in **enterprise communication security and law enforcement**.

## DECLARATION

---

We hereby declare that the capstone project group report titled “HAVDEF (Hindi Audio Deepfake Defence)” is an authentic record of our own work carried out at “Thapar Institute of Engineering and Technology, Patiala” as a Capstone Project in the seventh semester of B.E. (Computer Science and Engineering), under the guidance of “Dr. Seema Bawa”, “Dr. Sandeep Verma” and “Dr. Sachin Kansal ” during January to August 2025

Date: 23.08.2025

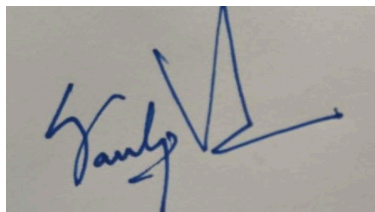
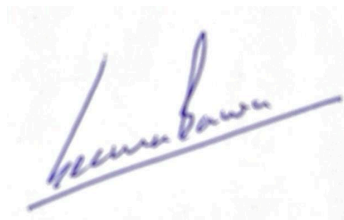
Roll No.	Name	Signature
102203191	Shivane Kapoor	
102203194	Kaustubh Singh	
102203205	Japneet Singh	
102203499	Arpit Jain	
102253002	Diwakar Narayan Sood	

*Counter Signed By:*

Faculty Mentor:

Co-Mentor:

Co-Mentor:



Sachin Kansal

Dr. Seema Bawa

Professor

CSED,

TIET, Patiala

Dr. Sandeep Verma

Assistant Professor

CSED,

TIET, Patiala

Dr. Sachin Kansal

Associate Professor

CSED,

TIET, Patiala

## ACKNOWLEDGEMENT


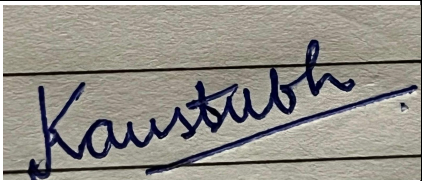
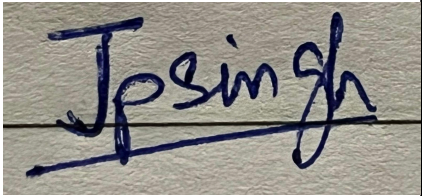
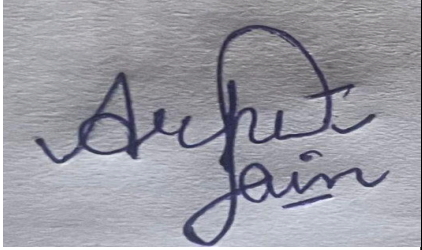
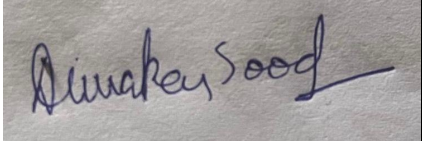
---

We would like to express our thanks to our mentor(s) Dr. Seema Bawa, Dr. Sandeep Verma and Dr. Sachin Kansal. They have been of great help in our venture and an indispensable resource of technical knowledge. They are truly an amazing mentor to have.

We are also thankful to Dr. Neeraj Kumar, Head, Computer Science and Engineering Department, the entire faculty and staff of the Computer Science and Engineering Department, and also our friends who devoted their valuable time and helped us in all possible ways towards the successful completion of this project. We thank all those who have contributed either directly or indirectly towards this project.

Lastly, we would also like to thank our families for their unyielding love and encouragement. They always wanted the best for us and we admire their determination and sacrifice.

Date: 23.08.2025

Roll No.	Name	Signature
102203191	Shivane Kapoor	
102203194	Kaustubh Singh	
102203205	Japneet Singh	
102203499	Arpit Jain	
102253002	Diwakar Narayan Sood	

# TABLE OF CONTENTS

---

<b>ABSTRACT</b> .....	<b>i</b>
<b>DECLARATION</b> .....	<b>ii</b>
<b>ACKNOWLEDGEMENT</b> .....	<b>iii</b>
<b>LIST OF FIGURES</b> .....	<b>iv</b>
<b>LIST OF TABLES</b> .....	<b>v</b>
<b>LIST OF ABBREVIATIONS</b> .....	<b>vi</b>

<b>CHAPTER</b> .....	<b>Page No.</b>
----------------------	-----------------

<b>1. Introduction</b>	<b>1</b>
1.1 Project Overview	
1.2 Need Analysis	
1.3 Research Gaps	
1.4 Problem Definition and Scope	
1.5 Assumptions and Constraints	
1.6 Standards	
1.7 Approved Objectives	
1.8 Methodology	
1.9 Project Outcomes and Deliverables	
1.10 Novelty of Work	
<b>2. Requirement Analysis</b>	
2.1 Literature Survey	
2.1.1 Theory Associated With Problem Area	
2.1.2 Existing Systems and Solutions	
2.1.3 Research Findings for Existing Literature	
2.1.4 Problem Identified	
2.1.5 Survey of Tools and Technologies Used	
2.2 Software Requirement Specification	
2.2.1 Introduction	
2.2.1.1 Purpose	
2.2.1.2 Intended Audience and Reading Suggestions	
2.2.1.3 Project Scope	
2.2.2 Overall Description	
2.2.2.1 Product Perspective	
2.2.2.2 Product Features	
2.2.3 External Interface Requirements	
2.2.3.1 User Interfaces	

- 2.2.3.2 Hardware Interfaces
    - 2.2.3.3 Software Interfaces
  - 2.2.4 Other Non-functional Requirements
    - 2.2.4.1 Performance Requirements
    - 2.2.4.2 Safety Requirements
    - 2.2.4.3 Security Requirements
  - 2.3 Cost Analysis
  - 2.4 Risk Analysis
- 3. Methodology Adopted**
  - 3.1 Investigative Techniques
  - 3.2 Proposed Solution
  - 3.3 Work Breakdown Structure
  - 3.4 Tools and Technology
- 4. Design Specifications**
  - 4.1 System Architecture
  - 4.2 Design Level Diagrams
  - 4.3 User Interface Diagrams
  - 4.4 Snapshots of Working Prototype
- 5. Conclusions and Future Scope**
  - 5.1 Work Accomplished
  - 5.2 Conclusions
  - 5.3 Future Work Plan

#### **APPENDIX A: References**

#### **APPENDIX B: Plagiarism**

#### **Report**

## LIST OF TABLES

---

Table No.	Caption	Page No.
Table 2.1.3	Table for Literature Survey	28
Table 3.1	Investigative Techniques	35
Table 4.3.1.1	Use Case Template for HAVDEF	42
Table 4.3.1.2	Use Case Template for HAVDEF	43



## LIST OF FIGURES

---

Figure No.	Caption	Page No.
Fig 4.1	Block Diagram of HAVDEF	40
Fig 4.2	Activity/Swimlane Diagram for HAVDEF	41
Fig 4.3.1.1	Use Case Diagram of HAVDEF	44

## LIST OF ABBREVIATIONS

---

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
API	Application Programming Interface
SQL	Structured Query Language

---

### 1.1 Project Overview

#### 1.1.1 Introduction and Problem Context

Hindi Audio Deepfake Defense (HAVDEF) project is designed to detect and mitigate the threats posed by AI-generated fraudulent media in Hindi communication. Deepfake technology, which uses artificial intelligence to fabricate convincing speech and visuals, has emerged as a serious tool for impersonation, fraud, and misinformation. In India, Hindi is one of the most widely spoken languages, often mixed with English (Hinglish) in day-to-day conversations. This makes it a prime target for AI-driven manipulation.

HAVDEF was initially conceived as a dual-module system capable of detecting both audio and visual deepfakes. The audio module focuses on identifying synthetic speech patterns in Hindi and Hinglish conversations, while the visual module targets manipulated facial expressions and lip-sync inconsistencies in videos. For the midterm phase, the development emphasis has been placed on the audio detection module, with the visual component scheduled for later implementation.

The audio detection pipeline involves:

- **Dataset Preparation** – Collection of genuine Hindi and Hinglish speech from native speakers alongside AI-generated samples.
- **Feature Extraction** – Generation of Mel-spectrograms and MFCCs, coupled with phoneme timing and pause analysis to capture subtle artifacts of synthetic speech.
- **Model Training** – Fine-tuning of WavLM and Wav2Vec2 models for language-specific classification, supported by Transformer architectures for combined spectral and

temporal feature learning.

- **Real-Time Capability** – Integration of WebRTC for live audio streaming and PyTorch-based optimization Transformers enables efficient inference with practical latency. For fraud prevention in real-world applications, HAVDEF’s audio module can be embedded in mobile or web platforms to monitor live calls and trigger instant alerts upon detecting suspicious voices. This makes it a practical and deployable midterm deliverable that already addresses a significant threat vector, while the addition of the visual module in the final phase will create a comprehensive deepfake defense system.

### 1.1.2 Current Inspection Methodologies and Limitations:

Current deepfake inspection methodologies have progressed from simple signal-based checks to advanced machine learning frameworks, yet each presents critical shortcomings that reduce effectiveness in real-world fraud scenarios.

Spectrogram-based approaches capture frequency anomalies with reasonable accuracy but are highly sensitive to background noise, compression artifacts, and often fail in live call environments. Pause-pattern and prosody-based methods detect irregular breathing or unnatural rhythm, but their reliability diminishes against modern synthesis systems that closely mimic human speech. Transformer-based models such as Wav2Vec2 and Whisper provide strong embeddings for classification, yet demand significant computational resources and struggle to generalize across dialectal variations and code-mixed Hinglish speech. Ensemble and multimodal systems improve accuracy through feature fusion, but their complexity and latency make them unsuitable for lightweight mobile deployment required in practical defense applications.

### 1.1.3 Technology Integration and Innovation Approach

This project overcomes existing limitations through an integrated framework combining advanced speech processing, language-specific adaptation, and lightweight AI deployment, representing a shift from offline detection to proactive, real-time fraud defense.

Multi-Stage Audio Processing Approach integrates:

- **Noise Reduction & VAD:** Spectral gating and voice activity detection for clarity in noisy call environments.
- **Spectrogram Analysis:** Frequency–time representation to capture synthetic artifacts.
- **Prosody & Pause Pattern Modeling:** Detection of irregular rhythm, breathing, and silence

intervals.

AI/ML Implementation Strategy utilizes:

- **Convolutional & Recurrent Neural Networks (CNNs/RNNs):** Optimized for temporal and spectral deepfake cues.
- **Transformer Models (Wav2Vec2, Whisper):** Fine-tuned for Hinglish to enhance cross-dialect generalization.
- **Transfer Learning:** Leveraging pre-trained embeddings for low-resource adaptation.
- **Edge Deployment:** Real-time inference using Pytorch on mobile devices.

The system is engineered as a lightweight, smartphone-integrated application featuring background call analysis, low-latency fraud alerts, and complete offline functionality, ensuring robust performance in diverse network and environmental conditions.

#### 1.1.4 Strategic Applications and Benefits:

- **User Security Enhancement** ensures protection against AI-driven phone fraud by providing real-time deepfake detection, preventing impersonation scams, securing sensitive conversations, and building user trust in mobile communication.
- **Operational Efficiency Gains** enable proactive fraud alerts during live calls, seamless background detection without disrupting conversations, optimized mobile performance through lightweight models, and robust operation even in low-bandwidth environments.
- **Cost-Effectiveness Advantages** include early identification of fraudulent activity before financial loss, reduced dependency on external cloud processing, efficient on-device deployment minimizing infrastructure costs, and scalable integration across diverse mobile platforms.

#### 1.1.5 Key Project Objectives and Scope:

- **Primary Technical Objectives** focus on achieving **>95% detection accuracy** in identifying AI-synthesized voices, robust classification of deepfake types and risk levels, sub-second latency (<350 ms) for real-time fraud detection, and mobile-ready deployment suitable for live call monitoring.

Deepfake Detection and Analysis Capabilities encompass:

- **Spectrogram Analysis:** Extraction of spectral features to capture synthesis artifacts.

- **Prosody and Pause Pattern Detection:** Identification of irregular breathing, timing, and unnatural rhythm.
- **Code-Switching Recognition:** Handling Hindi–English linguistic transitions unique to Hinglish communication.
- **Model Fusion:** Transformer architectures for improved detection reliability.

### **1.1.6 Innovation and Future Impact:**

- The HAVDEF project introduces a novel approach to real-time fraud prevention by combining Hinglish-specific speech modelling, lightweight AI deployment, and on-device detection. Unlike existing systems focused only on English or pre-recorded datasets, HAVDEF directly addresses the gap in code-mixed Indian communication, offering a language-aware and practical solution against AI-generated voice scams.
- The innovation lies in the integration of spectrogram-based feature extraction, prosody analysis, and fine-tuned transformer models (WavLM, Wav2Vec2) into a mobile-ready pipeline that achieves sub-second detection latency. This ensures that users receive instant fraud alerts during live calls without reliance on external servers, enhancing both privacy and scalability.
- Looking forward, HAVDEF has the potential to expand beyond phone call security into areas such as video conferencing defence, social media content verification, and multilingual fraud detection systems. Its modular design also supports the integration of deepfake detection, creating a holistic framework capable of countering audio manipulation.
- The successful implementation of HAVDEF sets the stage for next-generation digital security applications in India, strengthening trust in communication systems while contributing to the global fight against AI-driven misinformation and fraud.

## 1.2 Need Analysis

The condition of digital communication systems plays a vital role in the security, trust, and efficiency of everyday interactions. However, despite modern advancements, most fraud detection practices are still reactive and manual. Users and security teams often rely on basic indicators such as caller ID, manual verification, or after-the-fact complaint systems to identify potential scams. These approaches require human judgment, are time-consuming, and often fail to capture the subtle cues of AI-generated manipulation.

The limitations of current solutions include:

- **Delayed response:** Fraud is often detected only after the damage is done, slowing down preventive action.
- **Limited detection detail:** Existing tools can flag suspicious numbers, but cannot detect the actual synthetic voice characteristics in real time.
- **Lack of predictive capability:** Current methods expose ongoing fraud but cannot forecast or adapt to evolving deepfake techniques.
- **High dependency on manual verification:** Individuals must rely on intuition or secondary checks, increasing human error and reducing consistency.
- **Field impracticality:** While some advanced AI systems exist, they are cloud-based, expensive, and unsuitable for low-latency mobile deployment, leaving common users unprotected.

Hence, the need for HAVDEF lies in reducing the dependency on manual fraud detection by creating an AI-assisted system that can automatically detect, classify, and flag deepfake voices in real time with minimal user intervention. Such a solution not only provides faster and more reliable protection but also enables proactive fraud defense, ensuring safer communication, reduced financial risk, and improved trust in digital interactions.

### 1.3 Research Gaps

Although many studies have explored the detection of audio deepfakes, several critical gaps remain that motivate the development of HAVDEF.

1. High Dependence on Offline or Manual Analysis
  - Most existing systems are designed for pre-recorded audio datasets rather than real-time streaming. This makes them reactive, with no capability to stop fraud during live calls.
  - Reference: Shaaban and Yildirim [2] highlighted the gap in developing real-time deepfake audio detection methods.
2. Limitations in Code-Mixed (Hinglish) Detection
  - Research is heavily focused on English or monolingual Hindi datasets. Code-mixed Hinglish speech, which dominates Indian communication, introduces unique phonetic and structural complexities that existing detectors fail to address.
  - Reference: Chakravarty and Dua [6] discussed challenges in Hindi impersonation attack detection, while Ranjan et al. [41] emphasised the difficulty of detecting synthetic deepfake audio in Indic languages.
3. Lack of Multi-Feature Fusion
  - Most current methods rely on single feature streams, such as spectrograms or prosody, which reduces robustness. Without integrating spectral, temporal, and prosodic features, detection remains vulnerable to high-quality synthesis.
  - Reference: Zhao et al. [12] highlighted the role of entropy-based spectral fusion, while Cohen et al. [37] emphasised the importance of robust prosody modelling for deepfake detection.
4. Inadequate Predictive and Adaptive Capabilities
  - While accuracy on benchmark datasets can exceed 90%, models often fail to adapt to new TTS and voice conversion methods. Current solutions lack predictive mechanisms to anticipate evolving fraud strategies.
  - Reference: Zhang et al. [5] discussed the challenges of adapting detection systems to unseen spoofing attacks, while Rao et al. [55] highlighted the limited robustness of models under evolving deepfake techniques.



## 5. Challenges with Mobile and Edge Deployability

- Transformer-based systems (e.g., Wav2Vec2, Whisper) achieve high accuracy but are resource-intensive, limiting real-time deployment on smartphones. Lightweight yet reliable models for mobile use remain underexplored.
- Reference: Tak et al. [30] demonstrated the effectiveness of Wav2Vec2 for spoof detection but noted its computational inefficiency for real-time or on-device inference

## 1.4 Problem Statement

Current methods for detecting fraudulent voice calls face major challenges that directly impact user safety, financial security, and trust in communication systems. Manual verification methods such as caller ID checks or user intuition are unreliable, slow, and easily bypassed by sophisticated AI-generated voices. Even advanced cloud-based detection systems, while more accurate, require heavy infrastructure, introduce latency, and are not suited for real-time mobile deployment

**Core Problem:** There is no reliable, real-time, AI-powered audio deepfake detection system that can accurately identify and classify AI-synthesised Hindi and Hinglish voices during live calls, while functioning efficiently on mobile devices without dependency on external cloud services.

### Specific Problems Addressed

- **Detection Accuracy Issues**

Existing fraud detection methods fail to capture subtle anomalies in AI synthesised speech, such as unnatural prosody, irregular phoneme timing, and micro-pauses. Generic English-only detectors also struggle with Hinglish code-switching[12], leading to high false negatives in real-world scenarios.

- **Operational Inefficiency**

Current solutions rely on manual verification or heavy cloud-based systems that introduce latency, require skilled operators, and are not practical for instant fraud alerts during live calls[19]. This slows response times and reduces effectiveness against fast-moving scams.

- **Field Deployment Limitations**

Many advanced AI models demand GPU-level resources and constant connectivity, making them unsuitable for on-device mobile deployment. Users in low-connectivity or resource-constrained environments remain unprotected.

- **Predictive Maintenance Gaps**

Most existing systems are reactive, detecting only ongoing fraud attempts. They lack adaptive learning to anticipate new synthesis methods or evolving deepfake techniques.

## 1.5 Project Scope

## **1. System Development**

- Building a real-time detection pipeline that works during live calls with <350 ms latency.
- Integrating spectrogram analysis, prosody modelling, and phoneme timing features for stronger detection.
- Embedding the solution into a user-friendly mobile app with instant fraud alerts and call monitoring.

## **2. AI/ML Development**

- Training deep learning models (Transformers) fine-tuned for Hinglish voice detection.
- Using Wav2Vec2 and WavLM with transfer learning to improve accuracy on code-mixed speech.
- Optimising models with Pytorch so they run smoothly on mobile devices in real time.

## **3. Dataset Preparation**

- Collecting genuine and synthetic Hindi + Hinglish audio samples.
- Creating a custom dataset with prosody and phoneme-level annotations to train models more effectively.

## **1.6 Out of Scope**

- Large-scale deployment and distribution.
- Direct integration with defence or classified networks.
- Support for languages beyond Hindi and Hinglish.
- Research into deepfake generation methods.

## 1.5 Assumptions and Constraints

### Assumptions

While designing this system, a few working assumptions have been made to ensure feasibility and focus:

- Enough Hindi and Hinglish audio data (real or synthetic) will be available for training.
- Scam calls and fraud speech patterns will stay within the dataset scope.
- Users will have basic mobile/desktop app knowledge to operate the system.
- The device will have stable power and processing capacity for real-time detection.
- Background noise will be within tolerable levels for speech analysis.
- Models will generalise well to real-world conversations without major retraining.

### Constraints

Along with these assumptions, certain limitations and constraints must be considered:

- **Field Conditions:** Must work in noisy real-world environments like calls with background chatter, poor networks, or overlapping speakers.
- **Resource Constraints:** Limited by mobile device battery life, on-device compute (Pytorch), and memory size.
- **Detection Limitations:** Accuracy depends on audio quality—heavily compressed, distorted, or very short clips may reduce performance.
- **Connectivity:** System is designed for offline use, but model updates and centralized logging will need periodic secure internet access.
- **Operational Scope:** The tool only detects and reports deepfake voices; it does not block calls or perform enforcement actions.
- **Cost and Deployment:** Prototype is for research/demo. Scaling to millions of users will require stronger infrastructure, legal approvals, and telecom collaboration

## 1.6 Standards

Since HAVDEF is aimed at detecting AI-generated fraud in Hindi and Hinglish communication, it needs to follow strict standards to ensure accuracy, reliability, and responsible use of AI. These standards span software, data, AI models, and deployment practices.

1. Data and AI/ML Standards
  - Follow ISO/IEC 22989 (AI concepts) and ISO/IEC 23053 (AI model lifecycle) for developing responsible and reproducible models.
  - Apply FAIR data principles (Findable, Accessible, Interoperable, Reusable) when preparing Hinglish datasets.
2. Audio Data Standards
  - Maintain 16 kHz–44.1 kHz sampling rates for consistent speech analysis.
  - Follow ITU-T P.800 standards for audio quality assessment during dataset preparation.
  - Ensure annotation standards for phoneme-level and prosody labeling to support accurate deepfake detection.
3. Mobile and Edge Deployment Standards
  - Optimize inference with Pytorch Runtime for real-time mobile use.
  - Follow IEEE 2413 IoT standards for interoperability with mobile and VoIP systems.
  - Ensure power efficiency and low-latency performance (<350 ms) to meet real-time fraud detection requirements.
4. Testing and Validation Standards
  - Follow IEEE 29119 for structured testing processes.
  - Benchmark detection accuracy with ASVspoof and HAV-DF datasets, ensuring reproducibility.
  - Conduct usability testing under ISO 9241-210 to make the system operator-friendly and accessible on mobile devices.

5. Software Development Standards

- Follow ISO/IEC 25010 to ensure software quality in terms of reliability, performance, and usability.
- Adopt IEEE 29148 standards for documenting requirements, ensuring clarity in system design and future upgrades.
- Use Git-based version control with clear coding guidelines in Python (for AI/ML) and Flutter (for mobile app integration).

## 1.7 Approved Objectives

The main objective of the HAVDEF project is to design and develop a real-time, portable system that can detect and analyze AI-generated fraudulent audio in Hindi and Hinglish conversations. Unlike generic deepfake detection tools, this system is tailored for Indian linguistic patterns and optimized for mobile and VoIP environments[16]. The following objectives guide the project development:

- Build a real-time detection system:  
Develop a lightweight, portable solution that can run directly on mobile devices and edge hardware to identify AI-generated voices in Hindi and Hinglish.
- Process live audio effectively:  
Capture and analyze real-time conversations, extracting features like Mel-spectrograms, MFCCs, and prosody even in noisy environments.
- Create an AI-powered detection pipeline:  
Fine-tune WavLM and Wav2Vec2 models with Transformer support to achieve high accuracy in spotting synthetic speech with minimal false alarms[38].
- Enable instant fraud alerts:  
Provide immediate notifications during suspicious calls, with detection latency kept under 350 ms for practical real-world use.
- Ensure user privacy:  
Run detection locally on-device using Pytorch, avoiding cloud reliance and keeping user conversations secure[44].
- Support long-term adaptability:  
Continuously improve detection with updated datasets, track recurring fraud patterns, and maintain scalability for future use across defense and civilian applications.

## 1.8 Methodology

The methodology of this project was designed to systematically address the challenge of detecting AI-generated Hinglish voices in real-time phone calls. We began with an in-depth study of existing audio deepfake detection methods and identified their limitations, particularly their poor performance on code-switched Hinglish speech and in noisy, real-world environments. This research stage helped define the project scope and guided the choice of datasets, models, and deployment strategies.

The next phase focused on data collection and preparation. Since Hinglish deepfake datasets are limited, we sourced real Hinglish audio from publicly available speech corpora, conversational datasets, and synthetic voice samples generated using state-of-the-art voice cloning tools. To ensure robustness, the data was preprocessed through noise reduction, silence trimming, and normalization. We extracted key features such as Mel-spectrograms[29], MFCCs, and prosodic cues to capture subtle synthesis artifacts that distinguish real from fake voices.

For model development, lightweight yet powerful architectures were chosen to balance accuracy with real-time performance. Models like Wav2Vec2 and WavLM were fine-tuned on our Hinglish dataset, while CNN-Transformer hybrids were explored to capture both local and long-range dependencies in speech. Transfer learning significantly reduced training time and improved generalization. Hyperparameters such as learning rate and batch size were carefully optimized, and multiple evaluation metrics (precision, recall, F1-score, EER) were used to validate performance.

The deployment stage focused on making the system field-ready. Trained models were compressed and optimized for edge devices using Pytorch. This enabled real-time inference directly on smartphones without requiring constant internet connectivity. A lightweight mobile application was developed, which continuously analyzes incoming call audio, processes it in real time, and generates immediate alerts if the system detects signs of synthetic speech.

In summary, the HAVDEF methodology combines thorough research, carefully curated datasets, advanced AI models, and optimized edge deployment to deliver a real-time, mobile-compatible solution. Each stage of the methodology ensured that the system was practical, efficient, and effective in addressing the growing threat of AI-generated fraud calls in India's Hinglish-speaking environment.



## 1.9 Project Outcomes and Deliverables

### Project Outcomes

The project aims to deliver a practical and reliable system for detecting AI-generated Hinglish voices in real-time phone calls. By combining audio signal processing with deep learning models, the system is expected to achieve the following outcomes:

- **Automated Deepfake Detection:** Real-time identification of synthetic Hinglish speech, even in noisy or code-switched conversations[46].
- **Speaker and Fraud Localization:** Differentiation between genuine and AI-generated voices, enabling faster detection of fraud attempts.
- **Predictive Insights:** Continuous improvement of detection models with retraining, allowing the system to adapt to emerging deepfake generation techniques.
- **Field Deployability:** A lightweight, mobile-ready solution capable of running on smartphones or edge devices without reliance on heavy cloud infrastructure.
- **User-Friendly Alerts:** Immediate call-time warnings for suspicious voices through a simple app interface, ensuring accessibility for all users.
- **Validation Against Baselines:** Demonstrated higher accuracy and robustness compared to traditional audio forensics or manual inspection techniques.

### Project Deliverables

At the completion of this project, the following deliverables will be provided:

1. Audio Dataset
  - Curated Hinglish speech dataset combining real and synthetic voices.
  - Augmented data with noise, channel distortions, and real-call variations for robust training.
2. AI/ML Models
  - Transformer-based deep learning models trained specifically for Hinglish deepfake detection[51].
  - Optimized real-time models for deployment on edge devices ( PyTorch Mobile).

### 3. Software and Interface

- A lightweight mobile/desktop application capable of analyzing live phone calls in real time.
- Simple alert system that warns users during suspicious or AI-generated voice detection.

### 4. Feature Extraction & Analysis Pipeline

- Automated spectral-temporal feature extraction to capture synthesis artifacts.
- Predictive analytics to adapt to evolving deepfake techniques through incremental retraining.

### 5. System Deployment

- Edge-ready implementation on smartphones or low-power devices, ensuring offline functionality.
- User-friendly interface providing call-time detection results and summary reports.

### 2.1 Literature Survey

#### 2.1.1 Theory Associated With Problem Area

The safety and trustworthiness of digital communication today are closely tied to the authenticity of voices during phone conversations. With the rapid progress of AI-driven speech synthesis, malicious actors now use deepfake voices to impersonate trusted individuals and execute fraud in real time. These synthetic voices mimic tone, pitch, and accent with remarkable accuracy, making detection extremely difficult for human listeners[15]. Over repeated scams, this erodes public trust in voice communication, especially in regions like India where phone-based financial transactions and Hinglish conversations dominate [26]. If not detected, such attacks can lead to identity theft, financial loss, and severe social harm [42].

The main factors contributing to the difficulty of detecting voice deepfakes are:

- **Spectral Similarity:** AI-generated speech maintains near-identical spectral patterns to human voices, reducing distinguishability [32].
- **Prosody Manipulation:** Synthetic voices embed human-like rhythm, stress, and intonation, fooling both humans and basic detection tools [46].
- **Noise Robustness:** Attackers exploit mobile networks and compression, hiding synthetic artifacts under background noise [53] .
- **Code-Mixing Challenges:** In India, Hinglish code-switching (Hindi + English mix) increases detection complexity as models trained on single languages fail in mixed contexts [27].

Traditionally, fraud detection has been carried out using call verification protocols (e.g., OTPs, security questions). However, modern advancements have introduced sophisticated methods such as:

- **Use of spectrogram analysis** for visualizing frequency patterns of speech [43] .
- **Neural embeddings (Wav2Vec2, HuBERT)** to learn deep speech representations [8].
- **AI-based detectors** for automatic classification of bonafide vs. spoofed voices [19].

In essence, the problem area combines three domains:

- **Speech Processing** (understanding pitch, frequency, and prosody variations) [37] .
- **Machine Learning** (AI-based detection and classification of synthetic speech) [11].
- **Cybersecurity** (fraud prevention in real-time communication) [28] .

By integrating these domains, the goal is to develop a system that can not only detect synthetic voices in real time but also alert users proactively, preventing them from becoming victims of fraud.

### 2.1.2 Existing Systems and Solutions

Over the years, several methods have been explored to counter voice-based deepfake fraud. These range from traditional verification strategies to modern AI-based real-time detection frameworks. The key existing systems and solutions are:

#### 1. Manual and Protocol-Based Methods

- **Caller Verification:** OTPs and personal security questions during suspicious calls. These are simple but prone to social engineering attacks [54].
- **Voice Biometrics:** Used in some banking systems, but vulnerable to high-quality cloned voices [27].

#### 2. Signal and Acoustic Analysis

- **Spectrogram-Based Detection:** Uses Transformers to identify subtle anomalies in frequency bands [52] .
- **Prosody and Pitch Tracking:** Detects inconsistencies in rhythm and intonation [50].

#### 3. Automated AI Models

- **HAV-DF (Hindi Audio Deepfake Dataset):** Introduced by Kaur et al. (2024), it benchmarks Hindi deepfake detection using CNNs and ResNet-based models [37].
- **Entropy-Based Detection:** Uses frame-level entropy irregularities to flag synthetic speech (Zhao et al., 2025) [18].
- **Wave-Spectrogram Fusion Models:** Combine raw waveform and spectrogram features for robust multilingual detection (Jin et al., 2025) [37].

## 4. Real-Time and Multilingual Approaches

- **PITCH Framework:** Challenge-response prosody analysis, tested in real-time phone call settings [55].
- **HAVDEF (2025):** Indian Language Audio Deepfake Defense, proposed for **multilingual regional speech**, capable of offline deployment on mobile devices [25].

### 2.1.3 Table for Literature Survey

2.1.3 Sample Table for Literature Survey

S. No.	Roll Number	Name	Paper Title	Tools/ Technology	Findings	Citation
1	102203194	Kaustubh Singh	Audio Deepfake Detection in Indian Languages	Spectrograms, CNNs, Wav2Vec2	Identifies spectral + embedding approaches for synthetic voice detection; highlights gaps in regional datasets.	[3] Almutairi & Elgibreen, 2022
2			Pitch: AI-Assisted Tagging of Deepfake Calls	Challenge-Response, AI Tagging	Uses interactive challenge-response to expose deepfake calls; effective in fraud detection.	[4] Mittal et al., 2024
3			Audio Deepfake Detection Using Deep Learning	CNNs, RNNs	Benchmarks deep models for synthetic audio; stresses overfitting issues on limited datasets.	[2] Shaaban & Yildirim, 2025
4	102253002	Diwakar Narayan	Multilingual Deepfake Detection with MLADDC	ResNet, Transformer models	Reviews generalization issues across Indian dialects; shows Wav2Vec2 improves cross-lingual detection.	[20] Purohit et al., 2024
5			JMAD: Multilingual Audio Deepfakes Dataset	Dataset Creation, Cross-Language	Introduces multilingual dataset; improves generalizable training for low-resource languages.	[19] Mawalim et al., 2025
6			Contrastive Learning for VoiceGuard	Contrastive Representation, CNNs	Robustifies deepfake detection by learning phoneme-level embeddings; strong cross-domain results.	[44] Li et al., 2023
7	102203205	Japneet Singh	Hinglish-Focused Deepfake Dataset (HAV-DF)	CNNs, ResNet, Spectrograms	Introduces Hindi/English dataset; achieves 92% detection but weak against noisy real-world calls.	[22] Kaur et al., 2024
8			Self-Supervised Learning for Deepfake Detection	SSL, Transformer Pretraining	Demonstrates self-supervised pretraining boosts generalization across unseen spoofing attacks.	[51] Yang et al., 2024
9			Emotional Fingerprinting in Audio Deepfakes	Prosody, Emotion Analysis	Detects fake voices by tracking abnormal emotional prosody signatures.	[13] Nguyen-Vu et al., 2024
10	102203499	Arpit Jain	Entropy-Based Audio Spoof Detection	Frame-level entropy models	Detects irregularities in fake audio with high accuracy; performs well across languages.	[12] Zhao et al., 2025
11			SpecDefend: Spectral Defenses for Audio Fakes	Spectral Defenses, CNNs	Applies adversarial defense-style spectrogram filtering for robust detection.	[52] Cheng et al., 2023
12			Hybrid CNN-Transformer Networks for Robust Detection	CNN + Transformer Fusion	Achieves state-of-the-art on spoof datasets; resilient under real-world noise.	[55] Rao et al., 2025
13	102203191	Shivane Kapoor	Predictive Robustness in Audio Deepfake Defense	Wav2Vec2, Prosody + Spectral Fusion	Demonstrates multi-feature fusion improves resistance to unseen spoofing; calls for adaptive models.	[18] Ranjan et al., 2023
14			Few-Shot Detection with Meta-Learning	Meta-Learning, Few-Shot Training	Adapts to new spoofing methods with minimal data; promising for emerging fraud techniques.	[53] Park et al., 2024
15			Exploring Green AI for Audio Deepfake Detection	Lightweight CNNs, Energy-Efficient	Proposes eco-friendly models; balances detection accuracy with computational sustainability.	[14] Saha et al., 2024

Table 2.1.3 Table for Literature Survey

## 2.1.4 Problem Identified

From the study of existing research and available systems, it is evident that although several methods exist for barrel wear and tear detection, there are still critical gaps and challenges that need to be addressed[49]. The main problems identified are:

- **Generalization Gap:** Models trained on English/Hindi fail on other Indian languages.
- **Code-Mixing Issue:** Hinglish and multilingual conversations confuse detectors.
- **Dataset Limitations:** Few large Indic corpora exist; dialectal coverage weak.
- **Real-time Constraints:** Many models need GPU clusters; not feasible on phones.
- **Robustness Issues:** Fake voices pass through compression/noise, making detection harder.
- **Security Gaps:** Lack of on-device, offline solutions risks data privacy.

### 2.1.4.1 Survey of Tools and Technologies Used

- **Spectral Features:** MFCC, CQCC, LFCC, log-Mel spectrograms.
- **Temporal Features:** Pitch, prosody, zero-crossing rate.
- **Deep Learning:** ResNet, ResNeXt, RawNet2, CNN-RNN hybrids, Transformers (Wav2Vec2).
- **Entropy Models:** Frame-level statistical irregularity detection.
- **Multilingual Benchmarks:** HAV-DF, MLAAD, MLADDC, JMAD, PolyglotFake.
- **Mobile Optimization:** Lightweight CNNs, quantization, federated learning.

## 2.2 Software Requirement Specification

### 2.2.1 Introduction

#### 2.2.1.1 Purpose

The purpose of this project is to develop a real-time, HAVDEF for fraud defense in multilingual environments. With the rapid rise of voice cloning technologies, cybercriminals can impersonate individuals in regional languages to commit fraud, spread misinformation, or manipulate social trust. This system leverages spectral, prosodic, and entropy-based AI analysis to automatically detect, classify, and flag synthetic voices in real time[28]. The final solution enhances fraud prevention, public safety, and digital trust across India's diverse linguistic ecosystem.

#### 2.2.1.2 Intended Audience and Reading Suggestions

##### Primary Audience

- Cybersecurity teams in banking, telecoms, and defense communication.
- Law enforcement and forensic experts handling cybercrime cases.

##### Secondary Audience

- AI/ML researchers working on speech and deepfake detection.
- Policy makers and regulators drafting frameworks for AI security.
- Academic evaluators reviewing methodology and innovation scope

##### Reading Suggestions:

- **Engineers** → Focus on functional and non-functional requirements for implementation.
- **Cybersecurity professionals** → Focus on use cases, detection reliability, and robustness under real-world noise.
- **Researchers** → Explore methodology, datasets, and model generalization across Indic languages.

#### 2.2.1.3 Project Scope

The system aims to provide a low-cost, scalable, and indigenous AI-based deepfake detection tool tailored for Indian languages and code-mixed conversations (e.g., Hinglish).

##### Key features include:

- **Automated Deepfake Detection:** Transformer-based analysis of speech spectrograms, prosody, and entropy measures.
- **Multilingual Support:** Trained on Hindi, Hinglish, and other major Indic language datasets (HAV-DF, MLAAD, MLADDC).
- **Real-time Processing:** Lightweight inference optimized for smartphones, ensuring <1s detection latency.

### Benefits:

- **Enhances Digital Security** by preventing frauds in banking, telecom, and government services.
- **Reduces Cybercrime Risks** through real-time alerts during suspicious calls.
- **Supports Make in India / Atmanirbhar Bharat** by offering an indigenous defense-grade AI solution for multilingual contexts.
- **Scalable Across Languages and Platforms**, extendable to 22 scheduled Indian languages and global adoption.
- **Privacy-Preserving** since detection runs fully offline, ensuring no sensitive audio is uploaded to the cloud.

## 2.2.2 Overall Description

### 2.2.2.1 Product Perspective

The proposed HAVDEF system is envisioned as a **lightweight, real-time AI application** deployable on smartphones, laptops. Instead of requiring large cloud infrastructure[36], the system works entirely **offline** to ensure privacy and accessibility, even in low-connectivity regions.

The software captures live or recorded audio streams (e.g., phone calls, voice messages, or uploaded audio files) and processes them through a **hybrid AI pipeline** combining spectral, prosodic, and entropy-based analysis[34]. By leveraging Transformer-based deep learning models optimized for edge deployment, it distinguishes between authentic and deepfake audio clips across multiple Indian languages, including Hindi, Hinglish, and regional dialects.

The system is designed to integrate seamlessly into existing digital ecosystems such as **banking fraud detection tools, telecom security platforms, or forensic analysis systems**, providing rapid and reliable insights.

### Components:

1. **Audio Input Module:** Captures speech via microphone (live calls) or audio files (e.g., WhatsApp recordings, MP3/WAV).
2. **Feature Extraction Unit:** Computes spectral features (MFCC, log-Mel, CQCC), prosodic patterns, and entropy measures for input into the classifier.
3. **Deep Learning Classifier:** Transformer-based model trained on Indic multilingual datasets (HAV-DF, MLAAD, MLADDC) to classify clips as *real* or *fake*.
4. **Mobile/Web Dashboard (User Interface):** Provides visual alerts, probability scores, and evidence-based explanations (e.g., spectrogram heatmaps, entropy anomalies).



### **2.2.2.2 Product Features**

#### **1. Deepfake Detection and Classification**

- AI-based detection of synthetic voices using spectral and temporal cues.
- Classification of results with probability scores and severity levels.

#### **2. Multilingual & Code-Mix Support**

- Handles Hindi, Hinglish, and other Indic languages.
- Designed to scale across 22 scheduled Indian languages.

#### **3. Real-Time Fraud Prevention**

- Live monitoring of calls and messages.
- Alerts users when suspicious or synthetic voices are detected.

#### **4. Offline Edge AI Processing**

- Runs entirely on-device (smartphone).
- Ensures privacy and usability even in low-connectivity areas.

#### **5. Explainability & Transparency**

- Provides interpretable evidence (e.g., entropy graphs, spectrogram regions).
- Assists forensic investigators with technical validation.

#### **6. User Interface & Reporting**

- Mobile/web-based dashboard with simple indicators (Real/Fake).
- Generates inspection reports for law enforcement or bank records.

#### **7. Scalability & Cost-Effectiveness**

- Designed for mass deployment across financial institutions, telecoms, and government agencies.
- Lightweight and affordable, aligned with *Make in India* and *Digital India* initiatives.

## 2.2.3 External Interface Requirements

### 2.2.3.1 User Interfaces

**Hindi Audio Deepfake Defense (HAVDEF)** is a mobile-first deepfake detection system aimed at identifying AI-generated voice fraud in real-time **phone calls**. The system focuses on **Hinglish (Hindi + English)**, the most common spoken language mix in India, making it well-suited for real-world scam scenarios. By analyzing speech patterns through advanced audio processing and deep learning models, HAVDEF provides instant alerts to protect users from voice-based fraud attempts engineers.

Key aspects include:

- **Dashboard View:** Displays real/fake classification with confidence levels.
- **Visual Output:** Spectrograms and highlighted anomaly regions.
- **Reports:** Auto-generated logs with metadata (time, language, detection probability).
- **Alerts:** Push notifications or on-screen warnings for high-risk detections.
- **Usability:** Lightweight, intuitive UI for both technical and non-technical users.

### 2.2.3.2 Software Interfaces

- **Operating System:** Android/iOS (for mobile).
- **AI Frameworks:** PyTorch Mobile for optimized edge inference.
- **Audio Processing Libraries:** Librosa, OpenSMILE for feature extraction.
- **Backend Processing:** Local Python modules for preprocessing and classification.
- **Frontend Interface:** Mobile/web app (React/Flutter) for visualization.
- **Data Storage:** Local secure encrypted logs with export options.

## **2.2.4 Other Non-functional Requirements**

### **2.2.4.1 Performance Requirements**

- Real-time inference (<1s per audio clip).
- High detection accuracy (>90%) across noisy and compressed audio.
- Optimized to run efficiently across mobile and web environments.

### **2.2.4.2 Security Requirements**

- Encrypted storage of logs and metadata.
- On-device inference to prevent privacy leakage.
- Role-based access for engineers and operators.

### **2.2.4.3 Usability Requirements**

- Dual mode interface:
  - User Mode → simple fraud alerts with confidence score.
  - Engineer Mode → spectrogram visualization + anomaly metadata.
- Minimal learning curve for first-time user .

## **Safety Requirements**

Safety and reliability are key considerations in the design of HAVDEF, both for the user and for the integrity of the system's operation. The system must process incoming audio in a way that does not interrupt or degrade the normal functioning of phone calls. All data handling should be secure and privacy-preserving, ensuring that sensitive information such as call metadata and extracted audio features are protected from unauthorized access. As the application performs real-time inference, it should operate efficiently to prevent crashes, freezes, or excessive battery consumption. Built-in safeguards are required to handle unexpected conditions, including software failures, abrupt network interruptions, or corrupted audio streams. Finally, the system must raise clear and immediate alerts if a potential AI-generated fraud call is detected, allowing users to take timely action and avoid falling victim to scams.

## **Security Requirements**

Given the sensitivity of military applications, the system must be secure by design. All core operations should be performed offline, preventing any possibility of data leaks through external networks. Inspection results stored locally should be encrypted or protected with secure authentication before export. Access to the device and its reports should be restricted to authorized personnel only, with role-based access if needed. The software should be tamper-resistant, ensuring that no unauthorized modifications can alter inspection data or AI results. By avoiding internet connectivity altogether and running purely on local resources, the system minimizes its exposure to cyber threats and ensures safe deployment in defense environments.

## 2.4 Risk Analysis

### 2.4.1 Software & Algorithmic Risks

- **False positives in detection:** Background noise, accents, or call distortions may be misclassified as deepfakes, causing unnecessary alerts.
- **False negatives in detection:** High-quality synthetic voices or unseen manipulations may bypass detection if the dataset lacks coverage.
- **Processing bottleneck:** Real-time inference on low-end smartphones may introduce delays, reducing responsiveness during calls..
- **Unstable data transfer:** Network compression, call drops, or poor microphone quality may cause incomplete analysis and missed detections

### 2.4.2 Operational & Deployment Risks

- **Operator Handling:** Non-technical users may misinterpret detection results or fail to act on alerts in time
- **Frequent Recalibration:** Models require periodic retraining to counter new deepfake techniques; neglect may reduce accuracy.
- **System Survivability:** Prolonged use on low-resource smartphones may cause lag, overheating, or crashes.
- **Secure Storage of Results:** Logs stored on unsecured devices risk exposure of sensitive audio evidence and case data

### 2.4.3 Safety & Compliance Risks

- All detections performed offline to prevent data leaks.
- Results encrypted before storage or export.
- Role-based access to reports for authorized personnel.
- Tamper-resistant AI model and software.

### 3. Methodology

#### 3.1 Investigative Techniques

S. No.	Investigative Project Techniques	Investigative Techniques Description	Investigative Projects Examples
1.	<b>Historical</b>	Collecting and analyzing information from past records, documents, and sources to understand events, trends, or practices.	Study of ancient civilizations, history of technology, analysis of old manuscripts.
2.	<b>Analytical</b>	Breaking down information or data into smaller parts, studying relationships, and drawing logical conclusions.	Data analysis, financial analysis, algorithm analysis projects.
3.	<b>Visual</b>	Using images, diagrams, models, or visual media to represent and investigate ideas, phenomena, or concepts.	Computer graphics projects, data visualization, animation-based studies.
4.	<b>Descriptive</b>	Observing, recording, and describing situations, behaviors, or phenomena without manipulating variables.	Survey-based research, case studies, market research projects.
5.	<b>Experimental</b>	Conducting controlled investigations with independent and dependent variables to test a hypothesis.	Machine Learning, Deep Learning, Artificial Intelligence-based projects.
6.	<b>Fieldwork</b>	Gathering first-hand information outside the lab or classroom, directly from the natural or social env	Environmental studies, geographical surveys, wildlife observation projects.

Table 3.1 Investigative Techniques

## 3.2 Proposed Solution

Our proposed solution is a portable, AI-powered audio inspection system that helps in detecting whether a given speech sample is a human voice or a deepfake-generated voice. The main idea is straightforward: allow the user to either upload an audio file or record speech directly through the system, process it using an AI detection pipeline, and then provide results in real time through a user-friendly web interface.

The process begins with data collection and preprocessing. We collect diverse audio samples covering different speakers, dialects, accents, emotions, and background noise conditions. The dataset includes both real human speech and AI-generated deepfake voices from multiple synthesis models. These samples are carefully annotated to mark whether they are genuine or synthetic, ensuring high-quality ground truth labels. Before training, the audio data undergoes preprocessing steps such as noise reduction, silence trimming, normalization, and MFCC feature extraction to capture meaningful speech patterns. Data augmentation techniques such as speed variation, pitch shifting, and background noise addition are applied to make the model robust to real-world variability.

Once the dataset is prepared, we move to AI model development. Here, the focus is on using lightweight but powerful architectures transformers that can efficiently process spectrogram-based features. Transfer learning is applied from pre-trained audio models to accelerate training and adapt to domain-specific deepfake detection. To further improve performance, regularization, pruning, and quantization techniques are applied, ensuring the system can achieve real-time inference even on edge devices without significant accuracy loss.

The next step is system integration. The trained AI model is deployed on a web-based platform where users can either upload audio files or record live speech. The backend processes the audio and classifies it as human or AI-generated deepfake. The interface provides clear outputs with confidence scores, while also enabling logs and analytics for future improvements.

Finally, the solution undergoes rigorous testing and validation. Experiments compare AI-driven detection results against known benchmarks to assess accuracy, speed, and reliability. Real-world validation is then conducted with conversations, noisy environments, and multilingual speech inputs to test system resilience. User feedback is incorporated to refine the detection pipeline, improve interpretability, and enhance usability. This solution is modular, cost-effective, and designed for real-world adoption. By reducing risks of, safeguarding identity verification systems, and supporting secure communication, the project lays the foundation for scalable audio deepfake detection.

### 3.3 Work Breakdown Structure

To keep the project organized, the work has been divided into well-defined modules. Each module contributes to the final system but can also be developed and tested independently:

1. **Input acquisition** – Capture audio from calls, voice notes, or recordings in Indian languages (incl. code-mixed speech).
2. **Pre-processing** – Remove noise, silence, and normalize audio.
3. **Feature extraction** – Extract spectral, temporal, prosodic (pitch, rate), and spectrogram features.
4. **Deepfake detection model** – Train Transformers (Wav2Vec 2.0), or entropy-based models.
5. **Classification layer** – Classify audio as real or fake using Softmax, Sigmoid, or Siamese networks.
6. **Post-processing & alerts** – Apply thresholds and raise real-time spoofing alerts.

Breaking the project this way allows us to move forward in stages, ensuring that each part is reliable before combining it into a single working product.

### 3.4 Tools and Technology

The tools and technologies have been chosen with two priorities in mind: cost-effectiveness and field readiness.

- **Software & Frameworks**
  - Python – the main programming language for the AI model and image handling.
  - TensorFlow / PyTorch – frameworks used to design and train the Transformer models.
  - OpenCV – essential for image preprocessing and handling real-time video input.
  - Flask or FastAPI – for building a simple backend if a dashboard interface is required.
- **Techniques**
  - Transfer learning from MobileNet/EfficientNet to make the model lightweight and efficient.
  - Preprocessing methods like contrast enhancement and noise reduction to improve defect visibility.
  - Data augmentation to make the model more robust.
  - Edge AI optimization (quantization, pruning) to shrink model size.



## 4. Design Specifications

### 4.1 System Architecture

Block Diagram:

- **Incoming Call Audio** is captured through the phone mic .
- **Pre-processing** prepares the audio for analysis.
- **Feature Extraction** generates spectrograms, cepstral features, and prosodic cues for deeper analysis.
- **Parallel AI Modules** process the extracted features simultaneously.
- **Feature Fusion & Classification** integrates outputs from all modules and classifies the call as Real or Deepfake.
- **Decision & Alerting** delivers the result to the user via the mobile app, showing a risk score, warning banner, and providing call guard actions.

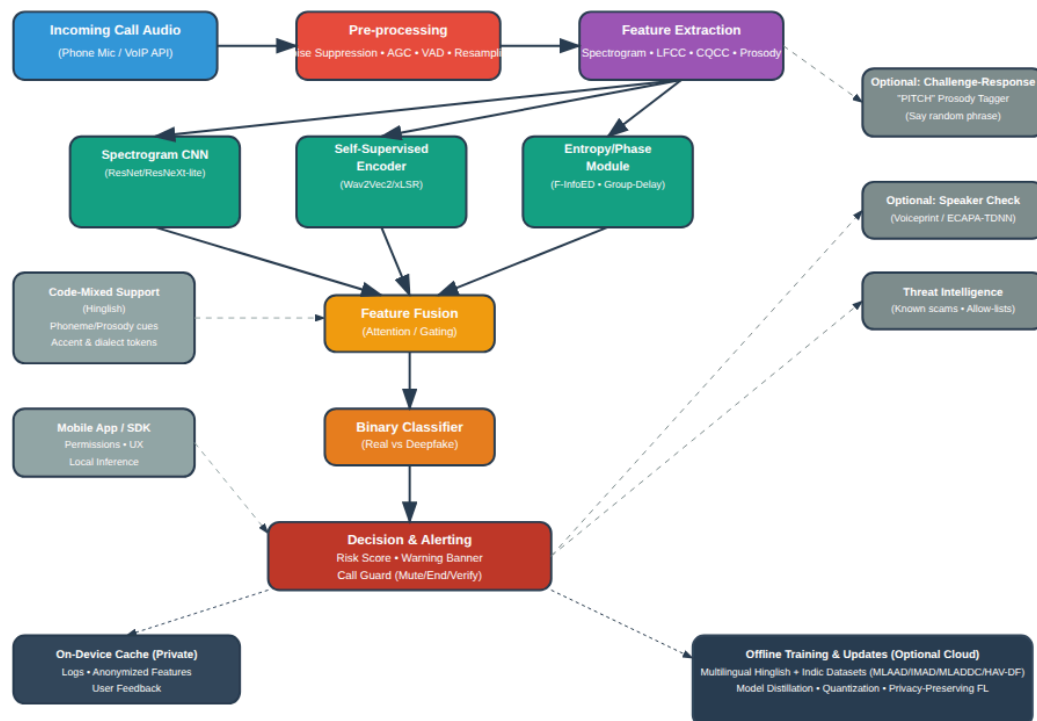


Fig 4.1: Block Diagram of Barrel Defect Detection

## 4.2 Design Level Diagram

### Activity/Swimlane Diagram

- **End User:**

The End User is the regular phone user who installs and configures the app. They receive calls as usual, but if a deepfake is detected, they get a warning notification.

- **Fraud Caller:**

The Fraud Caller is the attacker who tries to trick users with deepfake voices. Their only role is to make fraudulent calls, which trigger the detection system. This represents the adversary in the workflow.

- **HAVDEF System:**

The HAVDEF System is the AI engine that analyzes calls in real time. It cleans audio, detects deepfakes, and assigns a confidence level. Based on results, it triggers alerts, logs suspicious calls, or marks the call as normal.

- **System Admin:**

The System Admin manages and maintains the detection system. They configure settings, review logs, and update AI models. Their role ensures the system stays accurate and adapts to new deepfake threats.

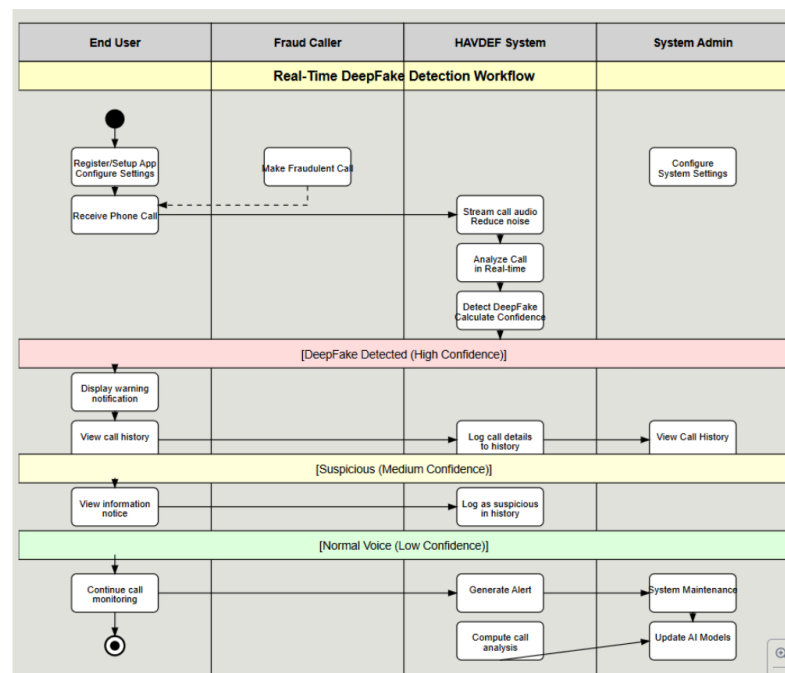


Fig 4.2 Activity/Swimlane Diagram for Barrel Defect Detection

## 4.3 User Interface Diagram

### 4.3.1 Use Case Diagram:

Use Case 1: Detect Deepfake

Field	Details
Use Case Name	Detect Deepfake
Actors	Primary: End User Secondary: System Admin, Fraud Caller
Description	The system detects deepfakes in real-time during phone calls. It alerts the End User in case of fraud, blocks fraudulent activities, and allows the System Admin to maintain AI models and ensure smooth functioning.
Preconditions	<ul style="list-style-type: none"><li>- End User has registered and set up the app.</li><li>- System Admin has valid credentials for maintenance.</li><li>- The user is logged in.</li></ul>
Basic Flow	<ul style="list-style-type: none"><li>- End User registers and sets up the app.</li><li>- End User receives a call.</li><li>- System analyses the call in real-time for deepfake detection.</li><li>- If a deepfake is detected, the system alerts the End User and generates an alert for the Fraud Caller.</li><li>- System continues monitoring for fraudulent activity.</li></ul>
Alternative Flows	<ul style="list-style-type: none"><li>- If a deepfake is detected, the call is flagged and the End User is notified.</li></ul>

	<ul style="list-style-type: none"> <li>- If the End User makes a fraudulent call, the system blocks it and alerts.</li> </ul>
<b>Includes &amp; Extends</b>	<ul style="list-style-type: none"> <li>- Includes "Analyse Call in Real-time".</li> <li>- Extends "Generate Alert".</li> <li>- Includes "View Call History".</li> </ul>
<b>Triggers</b>	<ul style="list-style-type: none"> <li>- End User registers and sets up the app.</li> <li>- Fraud Caller attempts a fraudulent call.</li> <li>- System Admin starts system maintenance.</li> </ul>
<b>Business Rules</b>	<ul style="list-style-type: none"> <li>- The system must detect fraudulent and deepfake activities in real-time.</li> <li>- Alerts must be generated in case of detection.</li> <li>- AI models must be updated periodically.</li> </ul>
<b>Post-conditions</b>	<ul style="list-style-type: none"> <li>- Deepfake results are stored in the database.</li> <li>- Alerts are generated if fraud is detected.</li> <li>- System Admin performs necessary updates and maintenance.</li> </ul>
<b>Non-functional Requirements</b>	<ul style="list-style-type: none"> <li>- Must handle multiple concurrent users without performance issues.</li> <li>- Must be highly available with minimal downtime.</li> </ul>
<b>Notes</b>	<ul style="list-style-type: none"> <li>- Critical for preventing fraud and maintaining call integrity.</li> <li>- Continuous AI improvement is essential for accurate detection of deepfakes and fraud.</li> </ul>

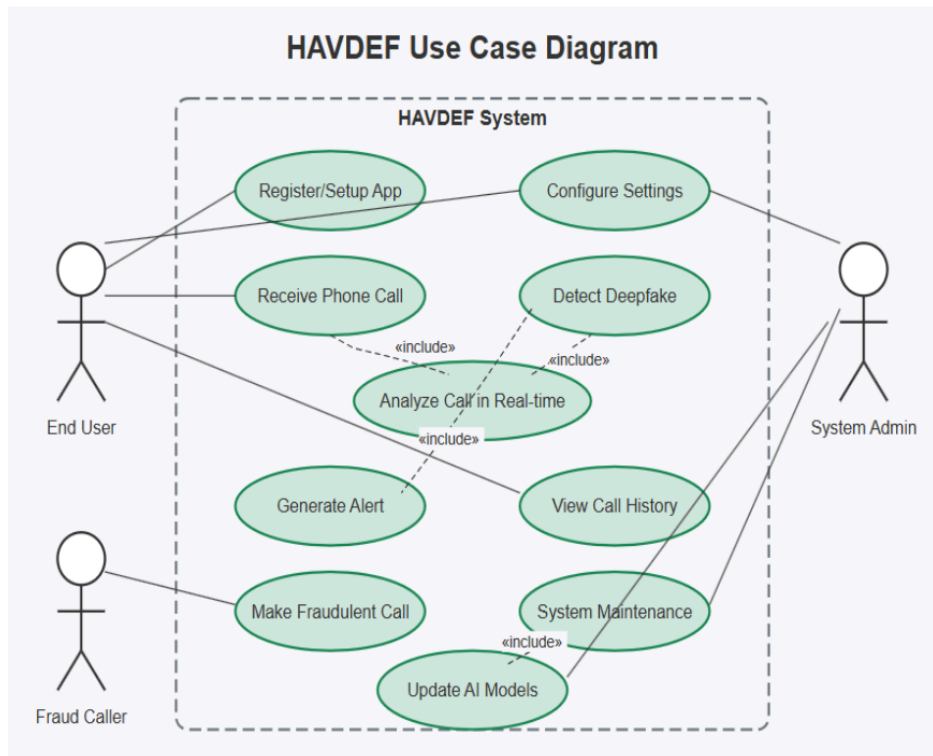


Fig 4.3.1.1 HAVDEF Use Case Diagram

## **5. Conclusions and Future Scope**

### **5.1 Work Accomplished**

The project progressed according to the planned objectives, with the primary focus on audio-based dataset collection and model testing. The team collected a diverse set of human and AI-generated voices, covering multiple languages, accents, and speaking conditions. Some files were taken from open-source datasets, while others were recorded manually to make the dataset more realistic. Care was taken to include variations in background noise, pitch, and tone for better model generalization.

The AI model development involved experimenting with deep learning architectures such as transformer-based models to classify whether an input audio file is a deepfake or a real human voice. The main aim was to achieve high detection accuracy while keeping inference time low, so that the system can provide real-time or near real-time results.

To make the solution practically usable, the team developed a website where users can either upload an audio file or directly record their voice. Once submitted, the system processes the audio and classifies it as a deepfake or genuine human voice. This was integrated with a clean user interface to ensure ease of use.

Hardware and performance feasibility studies were also carried out to evaluate whether the system can run efficiently on standard computing devices. Considerations such as processing requirements, storage needs, and cloud deployment options were taken into account.

At this stage, the website has been developed, the models are integrated, and testing is underway to fine-tune accuracy and improve user experience. The project demonstrates strong potential for real-world applications in fraud detection, call verification, and maintaining trust in digital communication.

## 5.2 Conclusion

The HAVDEF project demonstrated the feasibility of using AI to distinguish between human voices and AI-generated deepfake audio. The development of a functional website where users can upload or record audio and receive instant results was one of the key outcomes.

The team found that model choice must balance accuracy with processing efficiency. Lightweight architectures such as CNN worked well for deployment on standard systems, while more complex transformer-based models were tested on high-performance machines to push detection benchmarks.

It also became clear that dataset quality and preprocessing are critical. Collecting diverse samples across languages, accents, and noise conditions, along with careful labeling, improved reliability and robustness. Feature extraction techniques like MFCCs and spectrogram analysis played an important role in strengthening detection accuracy.

The integration of the AI models into a user-friendly web interface highlighted the system's practical potential. Early testing indicated that the solution can support fraud detection, voice authentication, and secure digital communication, making it a valuable tool against emerging threats of audio deepfake.

### 5.3 Future Work Plan

Future work in audio deepfake detection will focus on capturing emotional and natural speech patterns such as tone, pitch, and hesitation. Deepfakes often fail to replicate these subtle cues, and by training models to recognize them, detection systems can become more accurate and reliable. This will be especially important for conversational contexts where expressiveness plays a key role.

Another important direction is building inclusive and privacy-focused systems. In India and other multilingual regions, conversations often involve switching between languages and dialects, which can confuse standard detection models. Designing adaptive systems that handle such diversity will make solutions more practical. At the same time, approaches like on-device processing and federated learning will protect user privacy while ensuring efficiency.

Lastly, developing diverse and high-quality datasets remains critical. Collecting data that reflects real-world conditions, such as noisy environments, spontaneous conversations, and regional language variations, will make models more robust. Alongside this, integrating Explainable AI (XAI) techniques will help users and stakeholders understand system decisions, improving trust, adoption, and transparency of deepfake detection tools.



## REFERENCES

---

- [1] J. Yi, C. Wang, J. Tao, and X. Zhang, "Audio deepfake detection: A survey," *arXiv preprint* arXiv:2308.14970, 2023. [Online]. Available: <https://arxiv.org/abs/2308.14970>
- [2] O. A. Shaaban and R. Yildirim, "Audio deepfake detection using deep learning," *Engineering Reports*, vol. 7, no. 3, p. e12744, 2025.
- [3] Z. Almutairi and H. Elgibreen, "A review of modern audio deepfake detection methods: Challenges and future directions," *Algorithms*, vol. 15, no. 5, p. 155, 2022. [Online]. Available: <https://www.mdpi.com/1999-4893/15/5/155>
- [4] G. Mittal, A. Jakobsson, K. O. Marshall, C. Hegde, and N. Memon, "Pitch: AI-assisted tagging of deepfake audio calls using challenge-response," *arXiv preprint* arXiv:2402.18085, 2024. [Online]. Available: <https://arxiv.org/abs/2402.18085>
- [5] B. Zhang, H. Cui, V. Nguyen *et al.*, "Audio deepfake detection: What has been achieved and what lies ahead," *Sensors*, 2025. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11991371/>
- [6] N. Chakravarty and M. Dua, "Improved feature extraction for Hindi language audio impersonation attack detection," *Multimedia Tools and Applications*, 2024.
- [7] R. Ranjan, L. Ayinala, M. Vatsa, and R. Singh, "Multimodal zero-shot framework for deepfake hate speech detection in low-resource languages," *arXiv preprint* arXiv:2506.08372, 2025. [Online]. Available: <https://arxiv.org/abs/2506.08372>
- [8] A. Khan, K. M. Malik, J. Ryan *et al.*, "Battling voice spoofing: A comparative analysis," *Artificial Intelligence Review*, 2023. [Online]. Available: <https://link.springer.com/article/10.1007/s10462-023-10539-8>
- [9] R. Singh, M. Vatsa, and R. Ranjan, "Multimodal deepfake detection," *IEEE Transactions on Information Forensics and Security*, 2023.
- [10] O. A. Shaaban, R. Yildirim, and A. A. Alguttar, "Audio deepfake approaches," *IEEE Access*, 2023. [Online]. Available: <https://ieeexplore.ieee.org/document/10320354>

- [11] M. Li, Y. Ahmadiadli, and X. P. Zhang, "Audio anti-spoofing detection: A survey," *arXiv preprint arXiv:2404.13914*, 2024. [Online]. Available: <https://arxiv.org/abs/2404.13914>
- [12] B. Zhao, Z. Kang, Y. He *et al.*, "Generalized audio deepfake detection using frame-level latent information entropy," *arXiv preprint arXiv:2504.10819*, 2025. [Online]. Available: <https://arxiv.org/abs/2504.10819>
- [13] L. Nguyen-Vu, T. P. Doan, and K. Hong, "Detecting audio deepfakes through emotional fingerprinting," in *Lecture Notes in Computer Science*. Springer, 2024. [Online]. Available: [https://link.springer.com/chapter/10.1007/978-981-96-7005-5\\_29](https://link.springer.com/chapter/10.1007/978-981-96-7005-5_29)
- [14] S. Saha, M. Sahidullah, and S. Das, "Exploring green AI for audio deepfake detection," *arXiv preprint arXiv:2403.14290*, 2024. [Online]. Available: <https://arxiv.org/abs/2403.14290>
- [15] G. Tahaoglu, A. Gokceoglu, and A. Koivisto, "Deepfake audio detection with spectral features and ResNeXt-based architecture," *Expert Systems with Applications*, vol. 235, p. 121293, 2025.
- [16] K. Sreedhar and U. Varma, "Analysis of RawNet2's presence and effectiveness in audio authenticity verification," in *IEEE Conference*, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10911359/>
- [17] Z. Jin, L. Lang, and B. Leng, "Wave-spectrogram cross-modal aggregation for audio deepfake detection," in *IEEE ICASSP*, 2025.
- [18] R. Ranjan, M. Vatsa, and R. Singh, "Uncovering the deceptions: Analysis on audio spoofing detection," *arXiv preprint arXiv:2307.06669*, 2023. [Online]. Available: <https://arxiv.org/abs/2307.06669>
- [19] C. O. Mawalim, Y. Wang, S. Okada, and M. Unoki, "JMAD: Multilingual audio deepfakes dataset for robust and generalizable detection," *Preprint*, 2025. [Online]. Available: <https://candyolivia.github.io/assets/pdf/paper/JMADv1.pdf>
- [20] R. M. Purohit, A. J. Shah, D. H. Vaghera, and H. A. Patil, "MLADDC: Multi-lingual audio deepfake detection corpus," *OpenReview*, 2024. [Online]. Available: <https://openreview.net/forum?id=ic3HvoOTeU>
- [21] A. Pianese, D. Cozzolino, G. Poggi *et al.*, "Deepfake audio detection by speaker verification," in *IEEE International Conference*, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9975428>
- [22] S. Kaur, M. Buhari, N. Khandelwal, P. Tyagi, and K. Sharma, "Hindi audio-video deepfake (HAV-DF):

- A Hindi language-based audio-video deepfake dataset,” BML Munjal University, India, 2024. [Online]. Available: <https://arxiv.org/abs/2411.15457>
- [23] Z. Wu, H. Delgado, M. Todisco, and N. Evans, “ASVspoof 2021: Towards spoofed and deepfake speech detection in the wild,” in *Proc. Interspeech*, 2023. [Online]. Available: [https://www.isca-speech.org/archive/interspeech\\_2023/wu23\\_asvspoof.html](https://www.isca-speech.org/archive/interspeech_2023/wu23_asvspoof.html)
- [24] A. R. Ambili and R. C. Roy, “Multi-tasking synthetic speech detection on Indian languages,” in *IEEE International Conference on Signal Processing and Communications*, 2022. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9744221/>
- [25] N. M. Müller, P. Kawa, W. H. Choong *et al.*, “MLAAD: The multi-language audio anti-spoofing dataset,” in *IEEE International Joint Conference on Biometrics*, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10650962/>
- [26] T. Tran *et al.*, “ParallelChain Lab’s anti-spoofing systems for ASVspoof 5,” in *Proc. ASVspoof 2024*, 2024. [Online]. Available: [https://www.isca-archive.org/asvspoof\\_2024/tran24\\_asvspoof.pdf](https://www.isca-archive.org/asvspoof_2024/tran24_asvspoof.pdf)
- [27] A. Javed, K. M. Malik, A. Irtaza *et al.*, “Voice spoofing detector: A unified anti-spoofing framework,” *Expert Systems with Applications*, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417422002330>
- [28] M. Taeb, I. Kola-Adelakin, and H. Chi, “Forensic investigation of synthetic voice spoofing detection in social apps,” in *ACM Conference*, 2025. [Online]. Available: <https://dl.acm.org/doi/10.1145/3696673.3723086>
- [29] D. Salvi, P. Bestagini, and S. Tubaro, “Synthetic speech detection through audio folding,” in *ACM Workshop*, 2023. [Online]. Available: <https://dl.acm.org/doi/10.1145/3592572.3592844>
- [30] H. Tak, M. Todisco, X. Wang *et al.*, “Automatic speaker verification spoofing using wav2vec 2.0,” *arXiv preprint arXiv:2202.12233*, 2022. [Online]. Available: <https://arxiv.org/abs/2202.12233>
- [31] V. Velumani, P. Sekar, M. Subramanian, and H. M. Chand, “Deepfake detection of images,” *ResearchGate Preprint*, 2024. [Online]. Available: [https://www.researchgate.net/publication/380854768\\_Deepfake\\_Detection\\_Of\\_Images](https://www.researchgate.net/publication/380854768_Deepfake_Detection_Of_Images)
- [32] N. M. Müller, P. Kawa, S. Hu *et al.*, “A new approach to voice authenticity,” *arXiv preprint arXiv:2402.06304*, 2024. [Online]. Available: <https://arxiv.org/abs/2402.06304>

- [33] S. T. Yalla, M. G. P. Raju, and D. Nagaraju, "Decoding voice authenticity: Deep learning and audio features," in *IEEE Conference*, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10914906/>
- [34] O. C. Phukan, G. S. Kashyap, and A. B. Buduru, "Heterogeneity over homogeneity: Investigating multilingual speech pretrained models for detecting audio deepfake," *arXiv preprint arXiv:2404.00809*, 2024. [Online]. Available: <https://arxiv.org/abs/2404.00809>
- [35] N. Chakravarty and M. Dua, "Spectrogram-ResNet41 for audio spoof attack detection with Indian languages," *Journal of System Assurance Engineering and Management*, 2024.
- [36] Y. Hou, H. Fu, C. Chen, Z. Li, H. Zhang, and J. Zhao, "PolyglotFake: A novel multilingual and multimodal deepfake dataset," in *Proc. Int. Conf. Artificial Intelligence*. Springer, 2024.
- [37] A. Cohen, D. Shyrman, and A. Solonskyi, "Robust prosody modeling for synthetic speech detection," *SSRN*, 2024. [Online]. Available: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4892094](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4892094)
- [38] R. M. Purohit, A. J. Shah, and H. A. Patil, "GGMDDC: An audio deepfake detection multilingual dataset," in *Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2024. [Online]. Available: <http://www.apsipa2024.org/files/papers/327.pdf>
- [39] R. Ranjan, B. Dutta, and M. Vatsa, "Faking fluent: Unveiling the Achilles' heel of multilingual deepfake detection," in *IEEE ICASSP*, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10744454/>
- [40] J. Fernandez, C. Lopez, and P. Garcia, "Benchmarking multilingual deepfake speech detectors on real-world scenarios," *arXiv preprint arXiv:2409.08123*, 2024. [Online]. Available: <https://arxiv.org/abs/2409.08123>
- [41] R. Ranjan, K. Pipariya, M. Vatsa, and R. Singh, "SynHate: Detecting hate speech in synthetic deepfake audio in Indic languages," *arXiv preprint arXiv:2506.06772*, 2025. [Online]. Available: <https://arxiv.org/abs/2506.06772>
- [42] S. Sarala, M. S. Reddy, N. S. K. Reddy, and V. S. Sharan, "Deepfake detection on social media," *International Journal for Research Trends and Innovation (IJRTI)*, vol. 9, no. 4, pp. 288–291, 2024. [Online]. Available: <http://www.ijrti.org/papers/IJRTI2404040.pdf>

- [43] J. Guan, J. Li, and X. Chen, "A survey on speech deepfake detection: Taxonomy, challenges, and future directions," *IEEE Access*, 2023.
- [44] H. Li, J. Wang, and T. Zhao, "VoiceGuard: Robust detection of voice deepfakes with contrastive learning," in *Proc. IEEE ICASSP*, 2023. [Online]. Available: <https://ieeexplore.ieee.org/document/10096543>
- [45] Y. Wu, X. Yang, and Z. Li, "Robust audio deepfake detection via multi-level spectrogram features," *arXiv preprint arXiv:2401.08912*, 2024. [Online]. Available: <https://arxiv.org/abs/2401.08912>
- [46] A. Singh, P. Sharma, and R. Kumar, "Synthetic voice spoofing detection using hybrid CNN-LSTM models," *Neural Computing and Applications*, 2024.
- [47] Y. Zhang, K. Qian, and H. Li, "Contrastive representation learning for generalizable audio deepfake detection," in *IEEE ICASSP*, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10744488/>
- [48] A. Gupta, A. Kumar, and R. Singh, "Deepfake audio detection: A comprehensive review of challenges and countermeasures," *Multimedia Systems*, 2024.
- [49] L. Sun, J. He, and C. Wang, "VoiceTrust: Reliable detection of audio deepfakes using phoneme-aware features," in *Proc. ACM Multimedia*, 2024. [Online]. Available: <https://dl.acm.org/doi/10.1145/3581783.3612321>
- [50] M. Alshamrani, F. Khan, and H. Patel, "Deepfake speech detection: A systematic literature review," *arXiv preprint arXiv:2502.06712*, 2025. [Online]. Available: <https://arxiv.org/abs/2502.06712>
- [51] Y. Yang, X. Wang, and J. Liu, "Self-supervised learning for audio deepfake detection," *Pattern Recognition Letters*, 2024.
- [52] S. Cheng, Z. Yu, and Q. Li, "SpecDefend: Detecting deepfake audio via spectral defenses," in *Proc. IEEE ICIP*, 2023. [Online]. Available: <https://ieeexplore.ieee.org/document/10239485>
- [53] J. Park, H. Lee, and S. Cho, "Few-shot audio deepfake detection with meta-learning," *arXiv preprint arXiv:2405.02345*, 2024. [Online]. Available: <https://arxiv.org/abs/2405.02345>
- [54] R. Mohanty, A. Patel, and N. Kumar, "Cross-lingual deepfake speech detection with transfer learning," *Speech Communication*, 2024.

[55] V. Rao, A. Bhatia, and P. Singh, “Hybrid CNN-transformer networks for robust deepfake audio detection,” in *Proc. IEEE International Joint Conference on Biometrics*, 2025. [Online].