**FinTech-A-Thon ‘25**

**Team Name:** Friskel Freaks

**Problem Statement:** IDFC First Bank

**Team Members:**

Ajeethan

Karthikraj

Sharan Prashanth

Kishore

Vishwa Prasanth

**Core Challenge : AI-Driven Scam Call Detection**

**Tagline: “Why So Serious About Scams? Let’s End Them”**

**1. Problem Statement**

Scam calls are a growing threat, leading to financial fraud and privacy violations. Our goal is to build an AI-driven system that can screen, analyze and flag suspicious calls in real-time, ensuring that users are protected from potential scams.

**2. Idea / Proposed Solution**

**Overview**

Our system aims to alert scam calls using AI by integrating speech-to-text conversion, NLP-based scam detection, and risk scoring.

**How It Works**

1. **Speech-to-Text Conversion** – Converts call audio into text.
2. **Language Detection** – Identifies the language spoken during the call.
3. **NLP-Based Scam Detection** – Analyzes high-risk keywords & scam patterns.
4. **Caller ID Verification & Whitelisting** – Checks if the number is from a trusted source (e.g., banks).
5. **Risk Scoring** – Assigns a risk level based on detected fraud markers.
6. **User Alerts** – Real-time notifications (via SMS, app, email) for flagged calls.
7. **Continuous Learning** – Model improves over time based on user feedback.

**Innovation & Uniqueness**

AI-driven real-time scam detection.  
SMTP for instant alerts & call management.  
Combines multiple AI techniques for high accuracy.

**3. Technical Approach**

**Technologies Used**

* **Programming Language:** Python
* **Frameworks & Libraries:** TensorFlow, PyTorch, Transformers, Scikit-learn
* **APIs & Services:** Vosk Speech-to-Text Model, SMTP
* **Tools:** Jupyter Notebook, VS Code, GitHub

**Implementation Steps**

1️. **Setup & Install Dependencies**

* Python (≥3.8), TensorFlow, Transformers, VOSK model, SMTP.

2️. **Speech-to-Text Conversion**

* Used VOSK model Speech-to-Text API to transcribe calls.

3️. **Language Detection**

* Implemented langdetect to identify the call’s language.

4️. **NLP-Based Scam Detection**

* Trained a ML-based model to detect scam-related keywords.

5️. **Caller ID Verification & Risk Scoring**

* Compare with trusted database (e.g., bank numbers).
* Assign a risk score in the range(0-100) based on scam probability.

6️. **User Alert System**

* SMTP for SMS/email notifications when a high-risk call is detected.

7️. **Model Optimization & Testing**

* Fine-tune AI models to reduce false positives & false negatives.

**4. Feasibility & Viability**

**Feasibility Analysis**

**Technically Feasible** – Using existing AI & NLP models for accurate detection.  
**Economically Feasible** – VOSK model deployment.  
**Operationally Feasible** – Can integrate with mobile apps & telecom providers.

**Potential Challenges Faced & Solutions**

|  |  |
| --- | --- |
| **Challenges faced** | **Solution** |
| Background noise affecting transcription | Used noise reduction & AI-enhanced speech models. |
| High false positives in scam detection | Using a Trained model to detect this activity using various datasets |
| Latency in real-time analysis | Optimized speech processing for fast response time. |

**5. Impact & Benefits**

**Target Audience**

**Bank customers** – Preventing fraud from fake bank calls.  
**Elderly users** – Often targeted by scam calls.  
**Businesses** – Protecting employees from fraudulent calls.

**Key Benefits**

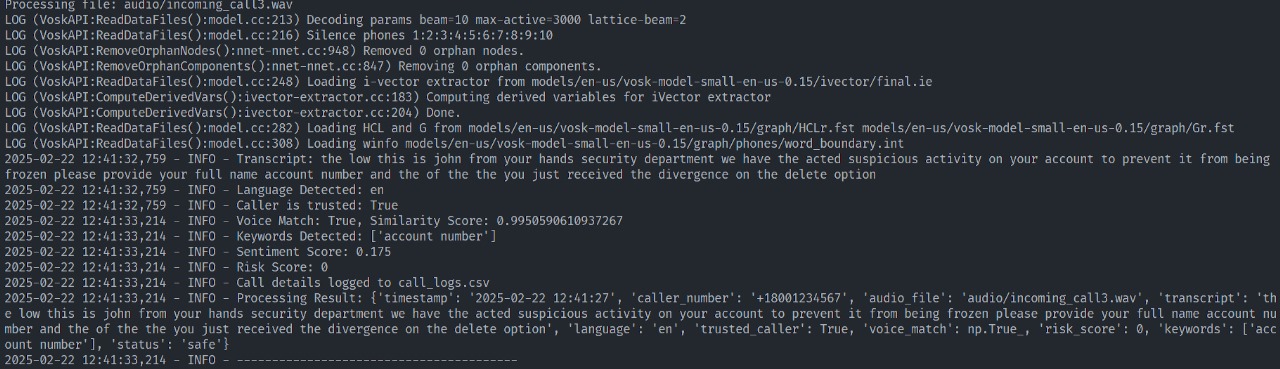
Prevents financial fraud & phishing scams.  
Enhances security for phone users.  
Empowers users by providing scam risk scores in real-time.

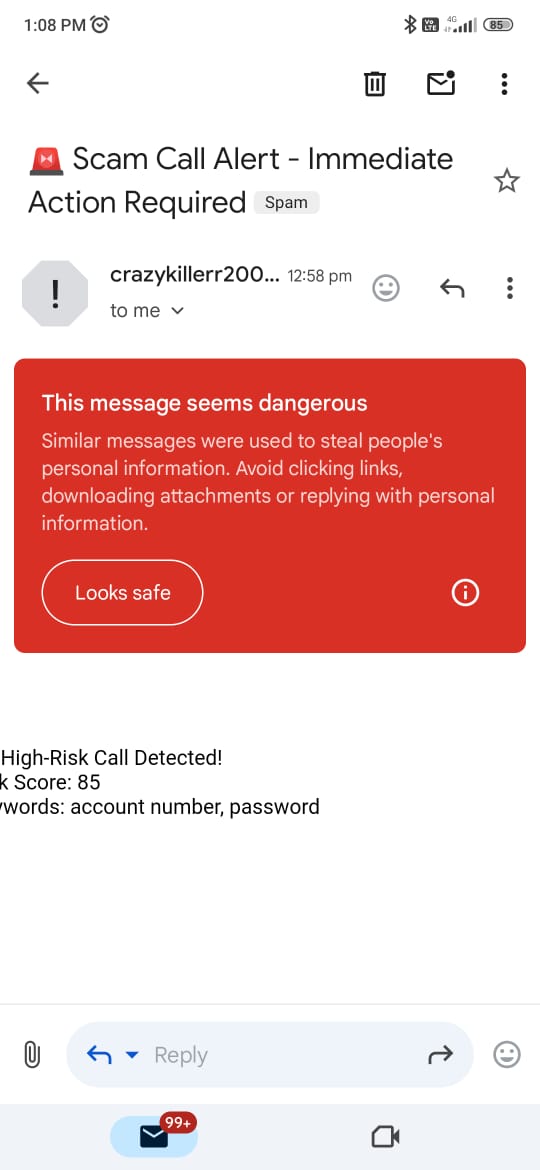
**6. Progress Timeline**

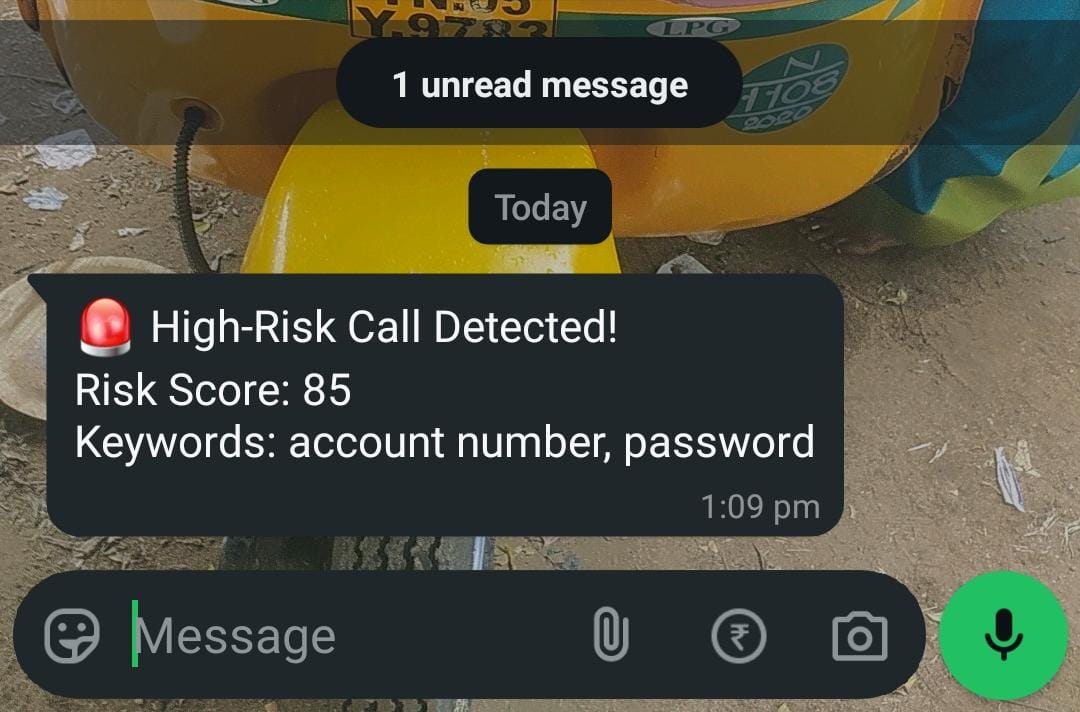
| **Step No.** | **Task** | **Description** | **Time Slot** |
| --- | --- | --- | --- |
| **1️** | **Environment Setup** | **Installed dependencies, configured APIs, and set up Python environment.** | **6:00 - 7:00 PM** |
| **2️** | **Speech-to-Text Conversion** | **Converted call audio into text using the VOSK model.** | **7:00 - 8:00 PM** |
| **3️** | **NLP Processing & Keyword Detection** | **Detects scam-related keywords.** | **8:00 - 9:30 PM** |
| **4️** | **Risk Scoring Algorithm Implementation** | **Assigns risk scores based on keyword patterns and ML models.** | **9:30 - 11:00 PM** |
| **5** | **Model Training** | **Model Training with datasets taken from Kaggle.** | **11:00 PM – 1:00 AM** |
| **5️** | **Caller ID Verification & Whitelisting** | **Verify caller authenticity using SMTP and maintain a whitelist.** | **4:00AM – 6:00AM** |
| **6️** | **Backend API Development** | **Developed FastAPI-based backend to handle scam detection requests.** | **6:00 AM – 7:30 AM** |
| **7️** | **Testing & Debugging** | **Check for false positives/negatives, optimize risk detection logic.** | **7:30 AM –**  **9:00 AM** |
| **8️** | **Documentation & Submission** | **Finalize the report, prepare presentation slides, and demo video.** | **10:00 AM - 1:00 PM** |

**GitHub repo :** [**Link**](https://github.com/Karthik751-MR/EchoEclipse)

**Result :**

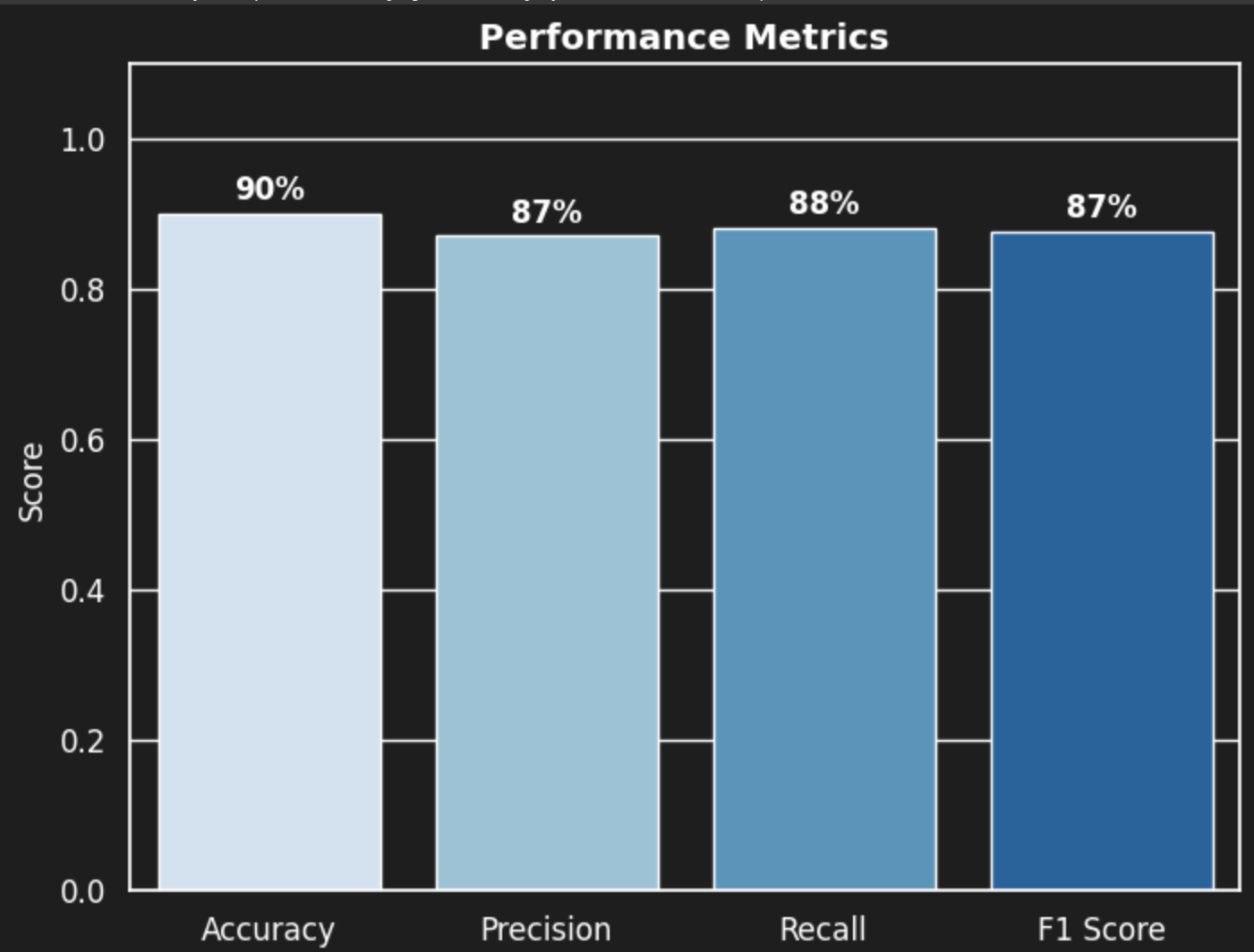






**Call\_logo\_file :** [**Link**](https://github.com/Karthik751-MR/EchoEclipse/blob/main/call_logs.csv)

**Performmace analysis :**

****

**Reference :**

1. **Vosk Speech-to-Text Model (Offline ASR Engine) – https://alphacephei.com/vosk**
2. **pydub (Audio Processing Library) –** [**https://github.com/jiaaro/pydub**](https://github.com/jiaaro/pydub)
3. **Hugging Face Transformers (NLP Models) – https://huggingface.co/docs/transformers/index**
4. **langdetect (Language Detection Library) –** [**https://pypi.org/project/langdetect/**](https://pypi.org/project/langdetect/)
5. **Scikit-learn (Machine Learning for Risk Scoring) – https://scikit-learn.org/stable/**
6. **SMTP (Simple Mail Transfer Protocol for Alerts) – https://www.rfc-editor.org/rfc/rfc5321**
7. **FastAPI (Backend Development) – https://fastapi.tiangolo.com/**
8. **Federal Trade Commission (FTC) – Phone Scam Awareness – https://consumer.ftc.gov/scam**