AI Enabled Scam Call Detection

Adwaith Anand, Arun Kumar A, Hariharan N, Harshavardhan A, Ishika Saxena, K J Rajendraprasad and Skanda Prasad H

Indian Institute of Science, Bengaluru

Abstract. The rise of digital communication has simultaneously enabled a surge in fraudulent messages and scam calls, threatening the privacy and security of users worldwide. In this work, we present a multimodal AI-driven fraud detection system capable of identifying scam content in both text messages and voice-based inputs, providing reasoning for its identification, and suggesting follow-up questions to the user. Our architecture integrates multiple components including Automatic Speech Recognition (ASR), Translation, Transformer-based classifiers, and Large Language Models (LLMs). We explore and evaluate four distinct approaches: (1) a fine-tuned Gemma-2 2.5B model, (2) a binary classification pipeline based on BERT, (3) a few-shot prompting + chain of thought strategy leveraging LLMs, and (4) a Retrieval-Augmented Generation (RAG) approach with a custom vector store. The choice of components used are made with future edge deployment in mind. To support robust experimentation, we augment the dataset using synthetic scam messages and call transcripts. We evaluate each of our approaches using classification metrics and analyze the trade-offs between latency, accuracy, and model size. Our results demonstrate that while transformer-based classifiers offer more speed and efficiency compared to RAG and few-shot LLM strategies which gives competitive edge in contextual understanding and reasoning. We deploy the end-to-end system via a web-based interface using Streamlit and FastAPI, enabling real-time analysis of both uploaded audio files, live recordings, and typed text.

1 Introduction

As digital communication channels continue to expand-from messaging platforms and SMS to voice over IP (VoIP) calls-scam and fraud attempts have become increasingly pervasive and sophisticated. Fraudsters often leverage psychological manipulation and social engineering to deceive victims. These attacks are not only growing in number but also in linguistic diversity, especially in multilingual regions like India.

Traditional fraud detection techniques, which rely on rule-based filtering or static keyword matching, fall short in addressing such dynamic and context-sensitive threats. These methods lack the flexibility to generalize across languages, adapt to new scam strategies and fail to serve populations that primarily communicate in Indic languages.

In this paper, we present a multilingual, AI-powered scam detection system designed for both text and voice inputs in Indic languages. Our architecture integrates ASR, translation, and LLM-based inference in a modular pipeline. It supports multiple inference modes-fine-tuned LLMs, few-shot prompting, and retrieval-augmented generation (RAG)-allowing it to adapt to varying compute budgets and deployment environments, including edge devices.

We conducted a comprehensive evaluation of these approaches using multiple criteria, including classification accuracy, precision, recall, F1 score, model size, inference latency, and compatibility with edge deployment. This analysis enabled us to understand the trade-offs between performance and efficiency for each strategy, and to identify the most practical configurations for real-world use across diverse deployment environments.

The key contributions of this work are:

1. A lightweight, modular architecture that integrates ASR, translation, transformer-based classification, and LLM inference.
2. A comparative evaluation of four fraud detection strategies, including fine-tuning, few-shot prompting with chain-of-thought reasoning, and RAG-based inference.
3. A curated dataset combining synthetic and real-world scam transcripts.
4. An open-source, real-time fraud detection application capable of handling both audio and text inputs interactively.
5. Practical guidelines for model selection-across LLMs, embeddings, ASR, and translation models-based on compute cost, time complexity, and deployment constraints.

2 System Architecture

2.1 Overview

Our system is designed as a multimodal fraud detection framework that supports both voice and text inputs through a web interface. The architecture (Figure 1) comprises three main components: *Input Interface*, *Preprocessing and Translation*, and *Inference and Classification*. Each component plays a key role in the overall pipeline for detecting fraudulent communication and generating intelligent follow-up suggestions.

2.2 Input Interface

The entry point of the system is a Streamlit-based web application that allows users to provide inputs in three forms:

1. Text Messages: Direct textual input, typically SMS or message logs.
2. Pre-recorded Audio Files: Uploaded by the user.
3. Live Voice Mode: Real-time audio recording through the browser interface.

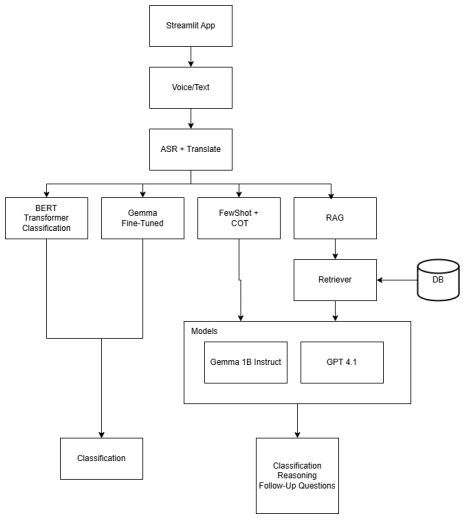

Figure 1. Architecture of the scam call detection system.

2.3 Speech-to-Text and Translation

For audio inputs, the system first performs automatic speech recognition (ASR) using a Conformer-based Indic ASR model[1]. The transcribed text or the direct text inputs is then passed to an English translation module. We use IndicTrans2-Distilled (200M)[2] for Indic-to-English translation. The translated text-now in a normalized English format-serves as input to the classification pipeline.

2.4 Classification and Inference

The translated text is evaluated through four independent fraud classification approaches:

1. **Encoder only BERT model**
2. **Gemma Instruction tuned**
3. **Few-shot + Chain of thought Prompting Approach**
4. **Retrieval-Augmented Generation (RAG) Approach**

Each approach returns:

- A binary classification: *Scam* or *Not Scam*
- A natural language explanation: *Classification Reasoning*
- A dynamic suggestion: *Follow-up Question* that can be posed to the caller to confirm or disprove suspicion

3 Methodology

To evaluate the different approaches, we generated a synthetic dataset containing text samples, corresponding labels, and category annotations. A diverse set of models was selected for each approach, based on factors such as computational cost, inference latency, and suitability for edge deployment. These models were then used in a series of experiments to assess their effectiveness and compare performance across the proposed strategies.

3.1 Model Selection

3.1.1 ASR Model

To enable speech-based scam detection across multiple Indic languages, we evaluated various automatic speech recognition (ASR) models. We generated a 100-audio-file dataset using synthetic call transcripts (fraudulent and legitimate) translated into Hindi, Tamil, and Kannada. These transcripts were synthesized using the Sarvam TTS engine to simulate real-world spoken call data as mentioned in B.

We benchmarked three multilingual ASR models on this dataset: IndicConformer-600M (IndicConformer-HMS), Google USM, and Meta MMS. Based on our results, IndicConformer-600M achieved the lowest average WER. Given its performance and open-source availability, it was chosen as the ASR backbone for our pipeline.

For detailed results and comparative analysis, refer to Appendix F.

3.1.2 Indic-English Translation Model

To handle multilingual inputs in Hindi, Tamil, and Kannada, we evaluated several translation models for their ability to accurately convert these languages into English.

We benchmarked three models - IndicTrans2-Distilled (200M)[2], NLLB-600M[5], and Google Translate - on this dataset using the chrF++ metric. Based on performance across all three languages, we selected IndicTrans2-Distilled due to its translation quality, efficiency, and open-source availability.

For detailed results and analysis of the evaluation, please refer to Appendix **??**.

3.1.3 LLM for Inference

In terms of Approach 3 and 4 we need an LLM model that is comparatively small in size, reduced time and compute complexity and a model that is suitable for edge deployment eventually.

We took 3 models that fit the above category namely Meta llama 3 1.2B IT, Gemma 3 1B IT and Qwen 1.5B IT, and further analyzed them in terms of parameters, int4 bit quantized model size, context window, MMLU-5-shot, Hellaswag-10shot, Arc-25 shot and edge friendliness. For detailed results and analysis of the evaluation, please refer to Appendix I

After analysis we concluded that Gemma 3 1B Instruct strikes balance in terms of accuracy and compute complexity for real time, privacy preserving scam detection on phones.

3.2 Synthetic Dataset Generation for Model Testing

A synthetic dataset is generated to evaluate a scam call and SMS classifier in the Indian context. This dataset simulates diverse communication scenarios to test the classifier's ability to distinguish fraudulent from normal communications. It includes text and multilingual audio data, covering various fraud and normal categories, scaled using large language models (LLMs), and tailored to Indian linguistic and cultural contexts. Detailed methodology is provided in Appendix B.

3.3 Approach 1: Supervised Fine-Tuning using Encoder-Only Model (BERT)

BERT was suitable for its significant contextual language ability, and this multilingual version was especially picked to allow text input in different languages and mixed languages, which are typical cases in scam messages. To adapt the bert-base-multilingual-cased model for a classification task of scam and non-scam call transcripts. The dataset was preprocessed, encoded, and divided into parts using stratification to keep the class balanced. The tokenization process involved cutting and filling up to 128 tokens in length. When cranking on a Kaggle T4 GPU with mixed precision (fp16) gave a

massive speed boost and enabled larger batches, the model ended up with an F1-score of 91.26%, with perfect precision (1.0) and recall of 83.93%, indicating that it was highly efficient in detecting the fraudulent transactions while it was still keeping false positives at a minimum which is a really important factor to avoid wrongly alarming genuine users.

Fine tuned model $\tilde{6}$82 MB) achieved an average inference time of 3.97 seconds per sample, thus making a practical balance between the accuracy and the usability in the real world. Detailed explanation available in Appendix **??**

3.4 Approach 2: Instruction-Tuned Decoder-Only Model using LoRA (Gemma 2B)

Gemma 2B is a decoder only model developed for text generation and summarization. To classify scam calls with model in generative nature, has been carried over with instruction-style prompting and the Low-Rank Adaptation (LoRA). LoRA-based fine-tuning was performed to adapt the large language model for specific tasks in an efficient way, using LoRA only 0. 07% parameters of the model (1.8M out of 2.5B) were changed. Especially, in the attention layers (q_proj, v_proj) where the adaptation to the task is most efficient. The k_proj layer was not modified in order to cut down on parts that are not really needed, since it plays a minor role in the specificity of the learning. The training took place on a T4 GPU because of the model's large size (9.5 GB base), while the fine-tuned adapter is only 56 MB. The model caught 94.55% of the total harmful messages (recall), meaning it is very good at recognizing fraudulent. Detailed explanation available in AppendixJ

3.5 Approach 3: Few Shot + Chain of thought Prompting

The input text received after preprocessing, ASR and translation is combined with a rich, self-contained prompt that has both few shot and chain of thought techniques embedded in it, with several worked examples, each showing a caller/text , a multi-step thought process and the final label , the reasoning followed up by follow up questions. Both this system prompt and user prompt is fed to Gemma 3 1B it model and is instructed to reply in a strict JSON schema.

Since Gemma lacks built in function-calling, an added JSON template system message compensates, its instruction tuned weights respond strongly to demonstrated reasoning patterns, improving factual coherence without needing temperature tricks.

3.6 Approach 4: RAG with Gemma

In terms of compute complexity this marks the most complex out of the prior three. It has two phases namely the Ingestion phase and the Retrieval phase. The ingestion phase is a one time process where 10k+ synthetically generated supervised dataset is preprocessed, chunked row wise with class and category as the metadata and embedded using BGE model. Once embedded, the chunks are stored in a vector DB.

In retrieval phase, when caller transcript is received post ASR and translation, it is used by retriever to fetch top-5 embeddings using dense similarity. Now along with this retrieved transcripts user transcript is passed to Gemma with a curated system prompt + user prompt combination. In depth workflow and model choice is defined in Appendix K

Similar to approach 3, JSON template response is extracted from LLM output and parsed for final classification result.

Table 1. Comparison of Model Approaches on F1 Score and Edge Deployment Suitability

Approach Name	**No. of Parameters**	**Context Limit**	**F1 Score**	**Edge Deployment Friendly (1–5)**
BERT	110M	512	0.91	4
Gemma LoRA	2.5B	8192	0.75	2
Few-Shot + CoT Gemma	1B	32k	0.59	3
RAG Gemma	1B	32k	0.48	2
Few-Shot + CoT GPT	1.8T	1M	0.98	1
RAG GPT	1.8T	1M	0.98	1

Note: Scale 1 = Low Edge Deployment Suitability, 5 = High.

Note: Full Table available at Appendix.

4 Evaluation And Results

We evaluate the system using standard classification metrics: All 6 approaches namely, fine tuned BERT Transformer, fine tuned Gemma llm, Few-shot+CoT Gemma, RAG Gemma, Few-shot + CoT GPT, RAG GPT was evaluated with a curated test dataset containing labels with 500 sample rows on precision, recall, accuracy, F1-score and average response time (subjected to vary due to different compute).

GPT approaches were also experimented and evaluated so as to explore cloud deployment solution as well, as in the call transcript would be encrypted and sent to gpt model llm in real time and relevant classification would be fetched. Indepth Evaluation details available in Appendix H

Along with this, they were also evaluated in terms of model size, number of parameters, context length, deployment friendliness. The final evaluation table is shown below.

5 Conclusion

Compared to the other approaches described in the paper, Approach 1 i.e. the fine-tuned BERT-based classifier offers the most balanced trade-off between performance and deployment feasibility, this is especially true when it comes to edge devices and low resource settings. This model is light-weight with only about 110 million parameters, which greatly reduces memory usage as well as power consumption. Thus, it can be used for implementation in smart phones or any other embedded system where resources are very limited.

Unlike RAG or Few-shot prompts, which are more complex, method 1 has less inference latency — meaning that it provides quick results without requiring a huge amount of compute power. It is also not grounded in external retrievals, long prompts, nor large-scale context which is a way to make the architecture simple and minimize the failure points.

6 Future Work

Going ahead, we have plans to further improve the system so that it can be practically used in the real world, especially edge environments. This also means that we will run the entire pipeline — ASR, translation, and classification — on low-resource devices like smartphones and embedded systems, thus allowing us to perform fraud detection in real time without relying on the cloud.

References

[1] ai4bharat. indic-conformer-600m-multilingual, 2024. URL https://huggingface.co/ai4bharat/indic-conformer-600m-multilingual. Accessed: 2025-06-18.

[2] J. Gala, P. A. Chitale, R. AK, V. Gumma, S. Doddapaneni, A. Kumar, J. Nawale, A. Sujatha, R. Puduppully, V. Raghavan, P. Kumar, M. M. Khapra, R. Dabre, and A. Kunchukuttan. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages, 2023. URL https://arxiv.org/abs/2305.16307.

[3] Narayanyadav. Fraud call india dataset, 2021. URL https://www.kaggle.com/datasets/narayanyadav/fraud-call-india-dataset. Accessed: 2025-06-18.

[4] U. M. L. Repository. Sms spam collection dataset, 2017. URL https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset. Accessed: 2025-06-18.

[5] N. Team, M. R. Costa-jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heafield, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard, A. Sun, S. Wang, G. Wenzek, A. Youngblood, B. Akula, L. Barrault, G. M. Gonzalez, P. Hansanti, J. Hoffman, S. Jarrett, K. R. Sadagopan, D. Rowe, S. Spruit, C. Tran, P. Andrews, N. F. Ayan, S. Bhosale, S. Edunov, A. Fan, C. Gao, V. Goswami, F. Guzmán, P. Koehn, A. Mourachko, C. Ropers, S. Saleem, H. Schwenk, and J. Wang. No language left behind: Scaling human-centered machine translation, 2022. URL https://arxiv.org/abs/2207.04672.

A Contributions

Adwaith Anand - ASR models evaluation and finalizing, Translation model evaluation and finalizing, evaluation of llama, Gemma and Qwen. ASR and translation pipeline Arun Kumar A - UI, Integration, App deployment, Devops, Base model evaluations. Hariharan - Dataset cleaning & handling, Model exploration and selection, Supervised Fine-Tuning using Encoder-Only Model (BERT), Instruction-Tuned Decoder-Only Model using LoRA (Gemma 2B), Evaluation metrics Harshavardhan A - AI Backend for Few-shot+CoT approach, RAG Approach, Evaluation of all approaches, Ishika Saxena - UI, Integration, Evaluation of models K J Rajendraprasad - Synthetic Dataset Preparation and Generation Skanda Prasad H - Synthetic Dataset Preparation and Generation, Evaluation Metrics

B Synthetic Dataset Generation

B.1 Category Identification

Analysis of online resources, including public datasets and industry reports, identified key categories for calls and SMS:

- **Fraud Categories**: Emotional Manipulation, Fake Delivery Scam, Financial Fraud, Identity Theft, Impersonation, Investment Scams, Job Offer Scam, Lottery Scam, Loan Scam, Phishing, Service Fraud, Subscription Scam, Tech Support Scam.
- **Normal Categories**: Delivery Update, Social, Service Inquiry, Entertainment, Work Update, Family, Sports, Recreation, Education, Travel.

These categories ensure comprehensive coverage of scam tactics and typical communications, enabling the classifier to detect contextual patterns and support targeted user alerts. The sample entries for each category from a generated dataset provided in Appendix D.

B.2 Reference Dataset Creation

The reference dataset was developed using Kaggle's "Fraud Call India Dataset" (Narayanyadav, 2021) [3] and "SMS Spam Collection Dataset" (UCI Machine Learning Repository, 2017) [4] as templates. These were extended via LLMs to include the identified categories, incorporating Indian-specific references. The fraud call dataset contains summarized transcripts with keywords and labels (category, fraud/normal), while the SMS dataset includes short messages with similar labeling. Manual curation ensured quality and diversity.

B.3 Large-Scale Dataset Generation

Automated Python scripts were developed to scale the dataset using LLMs. The scripts generate unique SMS messages and call transcripts, ensuring conversational realism and relevance to the Indian context. The process produces diverse entries, with mechanisms to maintain uniqueness and appropriate formatting, supporting robust testing of the classifier across varied scenarios. Sample prompts for generating call and SMS datasets are detailed in Appendix C.

B.4 Audio File Generation

To simulate real-world call scenarios, a script samples call transcripts and generates audio files using Sarvam text-to-speech API. Transcripts are translated into Hindi, Tamil, and Kannada (with English as a baseline) and converted to audio files. This multilingual audio dataset tests the classifier's performance on audio-based inputs, reflecting India's linguistic diversity.

B.5 Significance for Classifier Testing

The synthetic dataset provides:

1. **Diversity**: Covers multiple categories and languages, ensuring comprehensive testing.
2. **Scalability**: Enables generation of large datasets to evaluate model generalization.
3. **Realism**: Incorporates Indian-specific references and audio to mimic real-world conditions.

This dataset is critical for assessing the classifier's accuracy, robustness, and ability to handle multimodal (text/audio) and multilingual inputs, contributing to the project's goal of developing an effective scam detection system.

C Sample prompts for Dataset Generation

C.1 Overview

To support large-scale dataset generation for the scam call and SMS classifier, tailored prompts were designed for large language models (LLMs) to produce diverse, realistic call transcripts and SMS messages. These prompts ensure uniqueness, cultural relevance to the Indian context, and alignment with identified fraud and normal categories. Integrated into automated Python scripts, they enable scalable dataset creation with conversational realism. The following subsections detail the prompts for scam/normal call transcripts, scam SMS messages, and normal SMS messages.

C.2 Prompt for Scam Call Transcripts

This prompt generates concise, diverse call transcripts labeled as scam or normal, each tied to a specific category (e.g., Financial Fraud, Delivery Update). It uses reference dataset examples to guide the LLM, emphasizing unique wording, topics, and structures for varied classifier testing. The output includes the transcript and its category.

Here are some examples of scam call transcripts: Generate a new, short, diverse Indian scam call transcript in the same style belonging to one of the provided categories. Each example should be unique, use different topics, wording, and structure. Avoid repeating phrases or patterns. Also provide a suitable category for the call. "

C.3 Prompt for Scam SMS Messages

This prompt generates short fraud SMS messages (under 160 characters) for fraud categories (e.g., Phishing, Lottery Scam). It includes urgent or suspicious elements like fake URLs, phone numbers, or emotional triggers, mimicking real-world scam tactics. Indian-specific references and diverse tones (aggressive, threatening, manipulative) ensure variability and robustness.

Generate short fraud SMS messages (each under 160 characters). Include urgent, suspicious elements like fake URLs, phone numbers, or emotional triggers. Use unique, fictitious URLs. Include Indian-specific references. Messages should reflect a range of tones, including aggressive, threatening, or manipulative styles. Ensure each message is unique and varied. Return the messages as a numbered list. Base them on these examples:"

C.4 Significance of Prompts

The prompts enable a diverse, scalable, and realistic dataset:

1. **Diversity**: Varied wording, tones, and structures across categories and languages.
2. **Relevance**: Indian-specific references for cultural and contextual accuracy.
3. **Scalability**: Batch generation and example-based guidance for large-scale production.

These prompts, integrated into scripts, support comprehensive testing of the classifier's scam detection and normal communication differentiation capabilities.

C.5 Development | Testing Dataset Split

Using different prompts for generating synthetic data separately for development and testing is essential to ensure robust model evaluation and prevent data leakage. By designing distinct prompt templates for each phase, we create diverse datasets that minimize overlap in language, structure, and context. This approach helps simulate real-world scenarios more effectively, allowing the model to generalize better and reducing the risk of overfitting to specific patterns seen during development. Ultimately, it leads to more reliable assessment of model performance and enhances the credibility of testing outcomes.

D Samples of the Dataset

D.1 Significance

The "Samples of Dataset" provided showcase a collection of messages categorized as either "fraud" or "normal." The significance of this dataset lies in its potential use for Training / Testing machine learning models to detect fraudulent communications. Each entry is labeled, indicating whether it represents a fraudulent attempt (e.g., phishing, fake delivery, lottery scams) or a normal interaction (e.g., family messages, educational inquiries). By analyzing samples such as these, it helps better the understanding of the characteristics of fraudulent messages. This Synthetic dataset serves as a valuable resource for identifying patterns and trends in communication that may indicate fraudulent activity.

Table 2. Samples of the Dataset

Label	Category	Call_Transcript
fraud	Emotional Manipulation	Nani, it's your favorite grandson. I'm stranded in Goa with no money. Send Rs. 3,000 at `http://travelhelp-goa.in`. I'm desperate!
fraud	Fake Delivery Scam	Hello madam, this is from the courier company. Your package is stuck due to unpaid customs duty. Please pay Rupees 1,850 online to release it today.
fraud	Financial Fraud	Dear customer, we noticed some unusual activity on your bank account. To secure your funds, kindly share your debit card number and OTP immediately. Act fast or your account will be blocked.
fraud	Identity Theft	Dear customer, your PAN card is scheduled for suspension due to incomplete KYC. Kindly share your date of birth and full PAN number to avoid deactivation. Call 1800-258-XXXX now.
fraud	Impersonation	Hello, this is Ramesh from the Income Tax Department. There is a mismatch in your PAN details. To avoid penalty, kindly confirm your Aadhaar and bank account number now.
fraud	Investment Scams	Congratulations! You've been selected for an exclusive investment plan with double returns in 6 months. To get started, send your full name and PAN to this number now. Limited seats only!
fraud	Job Offer Scam	Namaste, this is Ramesh from HR department. For final round of your job interview at Infosys, we need a processing fee of Rs. 999. Please share your UPI details to proceed.
fraud	Loan Scam	Dear customer, you have been chosen for an exclusive SBI home loan offer at 2% interest! Please confirm your PAN and Aadhaar details to proceed.
fraud	Lottery Scam	Congratulations! Your mobile number has been randomly selected for a reward of Rupees 1,50,000. Kindly share your full name and bank details to process your winnings. Offer valid till today evening only.
fraud	Phishing	Immediate action required! Your Kotak account is compromised. Secure it: `http://kotak-safeaccess.in`
fraud	Service Fraud	Good afternoon, your electricity bill is pending due to a system error. To avoid immediate disconnection, please pay the outstanding amount now via UPI to securepower@upi.
fraud	Subscription Scam	Your Netflix subscription renewal failed. Please verify your card details at the link sent on SMS to avoid service interruption.
fraud	Tech Support Scam	Warning! Virus detected on your device. Secure it urgently by calling 1-800-333-9876 before data loss!
normal	Delivery Update	Just checked, your shoes arrived at the gate. Want me to keep them inside or leave at your door?
normal	Education	Did you submit the application for the campus fest, or should I handle it today?
normal	Entertainment	Heard you got the tickets for tonight! Let me know what time to meet at the gate.
normal	Family	Rohit, do you remember where we kept the electricity bill receipt? I need it for tomorrow's paperwork.
normal	Recreation	Will you be coming for badminton this evening or still working late?
normal	Service Inquiry	Just checked with the tailor, your suit will be ready by Friday evening. Do you want to pick it up or should I get it for you?
normal	Social	You coming to the Ganpati visarjan tomorrow? Let's go together!
normal	Sports	Hey, our school team qualified for the state finals! So happy!
normal	Travel	Nana, did you check if the train tickets got confirmed for next week?
normal	Work Update	Hey, I'm stuck in the metro-I'll reach office about 20 minutes late, okay?

E Few-Shot Learning and Chain-of-Thought (CoT) Prompt Sample

This appendix provides a few-shot learning and chain-of-thought (CoT) prompt sample used for classifying scam call transcripts. The prompt is designed to guide large language models (LLMs) in analyzing call transcripts, identifying scam patterns, and producing structured outputs for dataset generation. It is integrated into automated scripts for scalable dataset creation.

You are an AI Fraud/Scam Call Detector.

You will be given a transcript of what a caller (potential scammer) said during a phone call. Your task is to analyze this transcript and determine if it represents a scam call.

I'll provide you with several examples to help you understand different types of scam patterns. For each example, I'll show the thought process for classification.

EXAMPLES OF SCAMS AND THOUGHT PROCESSES:

Example 1: Phishing Transcript: "Dear customer, your Paytm wallet has been blocked for security reasons. Please confirm your PAN details to reactivate your account." Thought Process: This transcript has multiple red flags: 1. Creates urgency with account blocking 2. Asks for sensitive PAN details 3. Doesn't specify how the customer will be identified 4. Uses vague "Dear customer" greeting rather than person's name 5. No official verification channels mentioned Classification: Scam Confidence: 0.95 Reasoning: This is a clear phishing attempt targeting financial information. Legitimate companies don't ask for complete PAN details over calls. The urgency tactic is designed to prevent critical thinking. Follow-up questions: - "Can I call the official Paytm customer service to verify this issue?" - "What's the official Paytm helpline number I should be contacting?" - "How did you identify me as a customer without verifying my identity first?"

Example 2: Identity Theft Transcript: "Hello, this is Anjali from the Income Tax Department. There is an issue with your recent filing. Please provide your PAN number immediately to avoid penalties." Thought Process: 1. Claims to be from a government authority, creating fear 2. Mentions penalties to create urgency 3. Asks for sensitive information (PAN number) 4. No case number or official reference provided 5. No alternative official verification method offered Classification: Scam Confidence: 0.92 Reasoning: This is attempting identity theft. Government departments would send official notices first, provide reference numbers, and never ask for full PAN details over an unsolicited call. The pressure tactics are red flags. Follow-up questions: - "Can you provide your employee ID and department division?" - "What is the specific case/reference number for this issue?" - "Can I call back on the official Income Tax Department helpline to verify this matter?"

Example 3: Legitimate Call Transcript: "Hello, I'm calling from ABC Telecom about the network upgrade in your area. You might experience brief service interruption tomorrow between 2 to 4 PM. No action required from your side." Thought Process: 1. Provides specific company name and reason for calling 2. Only sharing information, not requesting any personal data 3. No urgency or pressure tactics 4. Gives specific timeframe 5. Explicitly states no action required Classification: Not Scam Confidence: 0.88 Reasoning: This appears to be a legitimate service notification from a telecom provider. It doesn't ask for any personal information, just informs about planned maintenance, and doesn't use high-pressure tactics.

When analyzing the new transcript, follow this chain of thought approach: 1. Look for common scam indicators (requests for personal/financial information, urgency, threats) 2. Analyze the legitimacy of the caller's claimed identity 3. Consider the nature of the request being made 4. Assess the pressure tactics or manipulation techniques 5. Draw conclusions based on comparison with known patterns

You MUST respond using the function $'classify_scam_call_schema'with the correct JSON structure.$

Be precise and helpful.

F ASR Model Evaluation

We evaluated three state-of-the-art multilingual ASR models on a 100-sample dataset consisting of synthetic fraud and non-fraud call audios in Hindi, Tamil, and Kannada languages. The audio files were generated using Sarvam TTS and aligned with text data used in our pipeline.

The models compared include:

- **IndicConformer-600M (IndicConformer-HMS)** – A high-performance, multilingual ASR model trained on Indian languages.
- **Google USM** – A commercial universal speech model supporting many languages, including Indic languages.
- **Meta MMS**(MMS-1B:FL102) – A multilingual ASR model released by Meta for 102 languages.

The Word Error Rate (WER) scores across the three languages are shown in Table 3.

Table 3. WER (%) for ASR models on synthetic 100-audio Indic dataset

Model	Hindi	Tamil	Kannada
IndicConformer-600M	**15.8**	**27.2**	**25.4**
Google USM	20.5	35.6	28.7
Meta MMS	23.1	31.4	26.9

The results indicate that IndicConformer-600M outperforms the other models across all three languages. Notably, it demonstrated substantial improvements in Tamil and Kannada, which are traditionally low-resource ASR languages. This makes it particularly suitable for deployment in multilingual Indian contexts where speech-based scam detection is required.

G Translation Model Evaluation

To quantitatively compare translation quality, we used a 100-row synthetic dataset containing call transcripts that reflect realistic fraud detection scenarios in Hindi, Tamil, and Kannada languages to form a multilingual evaluation set.

We evaluated three models:

- **IndicTrans2-Distilled (200M)** - A compact, open-source multilingual translation model optimized for Indian languages.
- **NLLB-600M** - Meta AI's open multilingual model trained on a large-scale corpus.
- **Google Translate** - Commercial translation engine widely used for general-purpose tasks.

The translations were evaluated using the chrF++ metric. The results are shown in Table 4.

As seen in the results, IndicTrans2-Distilled consistently outperformed the other models across all three languages. Its particularly strong performance on low-resource languages such as Tamil and Kannada makes it an ideal choice for the fraud detection pipeline targeting diverse Indic populations.

Table 4. chrF++ scores for translation from Indic languages to English (100-row dataset)

Model	Hindi	Tamil	Kannada
IndicTrans2-Distilled (200M)	**58.4**	**55.6**	**54.9**
NLLB-600M	53.9	50.8	49.7
Google Translate	56.2	52.1	51.4

H Evaluation Metrics in detail

In evaluating models for scam call / SMS detection, it is essential to use a range of metrics—accuracy, precision, recall, and F1 score—to gain a comprehensive understanding of each model's strengths and weaknesses. Accuracy provides an overall measure of how often the model is correct, but it can be misleading if the dataset is imbalanced. Precision indicates how many of the calls flagged as scams are actually scams, which is crucial to minimize false alarms. Recall measures how many actual scam calls are correctly identified, ensuring that the model does not miss potential threats. The F1 score balances precision and recall, offering a single metric that accounts for both false positives and false negatives, making it especially useful when the cost of errors is high.

H.1 Accuracy

The formula for accuracy is given by:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Meaning: Accuracy measures the overall correctness of the model, indicating the proportion of true results (both true positives and true negatives) among the total number of cases examined.

H.2 Precision

The formula for precision is:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Meaning: Precision indicates the proportion of true positive results in all positive predictions made by the model. It reflects how many of the predicted positive cases were actually positive.

H.3 Recall (Sensitivity)

The formula for recall is:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Meaning: Recall measures the ability of the model to find all the relevant cases (true positives). It indicates the proportion of actual positives that were correctly identified.

H.4 F1 Score

The formula for the F1 Score, which is the harmonic mean of precision and recall, is:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Meaning: The F1 Score provides a balance between precision and recall, especially useful when the class distribution is imbalanced. It is a single metric that captures both false positives and false negatives.

H.5 Definitions

- **TP (True Positives)**: The number of positive instances correctly predicted by the model.
- **TN (True Negatives)**: The number of negative instances correctly predicted by the model.
- **FP (False Positives)**: The number of negative instances incorrectly predicted as positive by the model.
- **FN (False Negatives)**: The number of positive instances incorrectly predicted as negative by the model.

H.6 Summary

- **Accuracy**: Overall correctness of the model.
- **Precision**: Correctness of positive predictions.
- **Recall**: Ability to find all positive cases.
- **F1 Score**: Balance between precision and recall.

Table 5. Performance Metrics of Different Approaches

Approach	Accuracy	Precision	Recall	F1 Score
BERT	0.91	1.00	0.83	0.91
Gemma LoRA	0.63	0.62	0.94	0.75
Few-Shot + CoT Gemma	0.48	0.92	0.48	0.59
RAG Gemma	0.60	0.77	0.60	0.48
Few-Shot + CoT GPT	0.98	1.00	0.98	0.98
RAG GPT	0.98	1.00	0.98	0.98

Based on the evaluation table, the Few-Shot + CoT GPT and RAG GPT approaches outperform others, achieving the highest scores across all metrics, including perfect precision and near-perfect recall and F1 scores. In contrast, Gemma LoRA and RAG Gemma show lower accuracy and F1 scores, indicating less consistent performance. BERT demonstrates strong performance, particularly in precision and F1 score, making it a reliable choice for edge deployment. By comparing these metrics, and taking into consideration Model Complexity and Ease of Deployment we conclude that BERT is the best performing model for AI Enabled Scam Call/SMS detection on edge devices.

I LLM Model Inference (Gemma)

Gemma 3, Llama 3 and Qwen 1.5 were analyzed on the below metrics and benchmarks, the metrics were finalized due to the following reasoning.

Parameter count/disk size – proxy for RAM footprint and number of MACs needed per decoded token, dominating edge latency and battery drain.

Context length – determines whether long scam transcripts or multi-turn interactions can be held in memory without windowing

Benchmark trio (MMLU, HellaSwag, ARC-Challenge) – covers factual reasoning (MMLU), commonsense continuation (HellaSwag), and science-style QA (ARC) - all skills a scam detector uses when classifying intent and drafting follow-up questions.

From table I it is evident that Gemma 3 1B-Instruct strikes the best accuracy-per-watt trade-off for real-time, privacy-preserving scam detection on phones: smallest RAM footprint, fastest decode, top commonsense score, permissive license. Llama-3 1.2 B is a solid fallback but huge in size compared to gemma, inference time is large and not optimized, and Qwen 1.8 B's extra parameters hurt latency without decisive quality gains.

Table 6. Comparison of Quantized Models for Edge Deployment

Model	Params	FP16 size*	Typical 4-bit size*	Context window	MMLU-5-shot / (Global-MMLU-Lite)	HellaSwag-10	ARC-25 shot	TPS (mobile NPU)	Quant. support	Edge-friendliness†
Gemma 3 1B-IT	1.0 B	≈2.0 GB	≈0.5 GB	32k	34.2	**62.3**	38.4	∼190 TPS (int4)	GGUF / GPTQ	**High**
Llama-3 1.2 B-IT	1.2 B	≈2.4 GB	≈0.6 GB	128k	**49.3**	41.2	**59.4**	∼183 TPS (FP8)	FP8 / GPTQ	Medium
Qwen 1.5 1.8 B-Chat	1.8 B	≈3.6 GB	≈0.9 GB	32k	46.7	61.4	37.9	∼120 TPS (int4)	GPTQ / AWQ	Low-Medium

J Task-Specific Fine-Tuning

J.1 Approach 1

We started with the bert-base-multilingual-cased model (110 million parameters) for a binary classification task of scam vs. normal call messages. This model was designed for multilingual text and has been successful in several classification tasks. The input dataset was artificially generated to simulate the features of real scam messages after which the nulls were removed, the labels were converted to binary integers, and the samples were distributed among the train and test sets in a stratified manner to ensure class balance. Tokenization was executed by employing the corresponding BERT tokenizer with truncation and zero-padding to a fixed-length string of 128 tokens in order to keep the batch processing uniform and efficient.

Hugging Face's Trainer API combined with the transformers library has enabled the training of a model on NVIDIA Tesla T4 GPU via Kaggle. The GPU setup significantly shortened training duration compared to local CPU and thus enabled the usage of batch size of 32, and mixed-precision (fp16) that increases throughput without losing accuracy. AdamW optimizer was employed and the model with the best F1-score was chosen as a representative of the dataset since the F1-score ensures a good balance between precision and recall. On the test set, the model has provided 91% accuracy, 100% precision, 83.93% recall, and an F1-score of 91.26%. High precision informs us that false positives (normal messages labeled as scam) were completely eliminated, which is very important as it will avoid unnecessary user alerts.

The rate of recall that is slightly lower means that some fraudulent messages may have been missed and this factor allows us to consider the possibility of using ensemble or multi-model strategies. The size of the final model was 682 MB, and the average latency per sample was 3.97 seconds, which makes the solution applicable to almost real-time usage in the detection of fraud scenarios.

J.2 Approach 2: Instruction-Tuned Decoder-Only Model using LoRA (Gemma 2B)

In our second approach, we fine-tuned Gemma 2B, a decoder-only LLM built for generative tasks, not classification. To adapt it to our binary classification problem, we used instruction-tuned prompts phrasing each input as "Classify this call: [text to be predicted] Label:" to let the model predict "fraud" or "normal" as a short response. Since full fine-tuning of Gemma's 2.5 billion parameters is resource-intensive, we applied LoRA (Low-Rank Adaptation), a technique under PEFT (Parameter-Efficient Fine-Tuning), which introduces small trainable modules in specific transformer layers.

We chose to apply LoRA only to the q_proj (query) and v_proj (value) attention sub-layers. These are known to be the most responsive to task-specific information: the query layer controls what the model is "asking" of the input, and the value layer determines what content is passed forward. In contrast, the k_proj (key) layer is more static and carries structural information about the sequence itself, rather than adapting directly to the target task.We left out k_proj on purpose to avoid unnecessary computational overhead and to make the training lighter and more focused. This design decision enabled us to get 94.55% recall by training only 0.07% (1.8M) parameters, while still having a model of a manageable size.

Training was done efficiently on a T4 GPU using fp16 mixed precision, and the adapter model was just 56MB in size. Although inference requires both the base model (9.5GB) and adapter, it still runs fast (1.58seconds/sample) and offers significant flexibility: the same base Gemma model can be reused across tasks with different small adapters, avoiding the cost of retraining from scratch.

K RAG Approach

The workflow of RAG approach is given in Figure 2

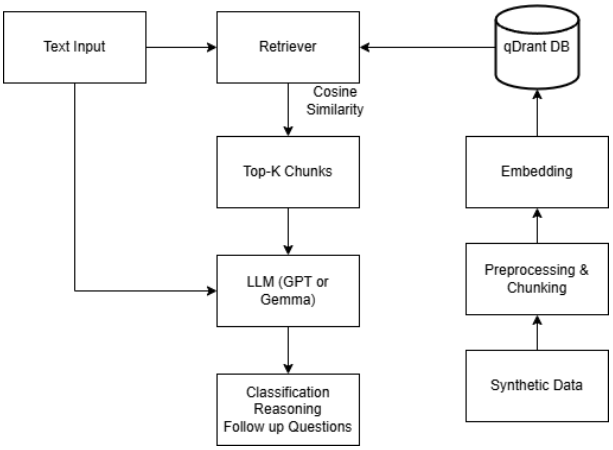

Figure 2. Workflow of RAG

K.0.1 Embedding Model:

Real-time scam detection needs a retriever that

1. Catches subtle paraphrases ("verify your account" "confirm KYC") to recall known scam scripts,
2. Runs inside phone-class hardware without draining the battery

In the Ingestion and retrieval phase, BGE model is used for embedding, especially BGE en v1.5, gives you near-state-of-the-art retrieval quality. fits in < 350 M parameters, is MIT-licensed, comes

pre-quantised/ONNX for mobile CPUs & NPUs. Those traits make it the most balanced choice for a scam-call RAG pipeline that must run both on-device and in the cloud.

BGE configuration is as follows:

- Architecture – 335M-Param BERT style Bi-encoder
- Vector Size – 1024 dims
- Max Input Length – 512 Tokens

In terms of Retrival quality with MTEB benchmark (Massive-Text Embedding Benchmark) BGE scores 64.2 avg across 56 tasks which is the current SOTA result.

K.0.2 Retrieval Strategy

For retrieval strategy there are several proven methods such as Dense retrieval, sparse retrieval, similarity search, Hybrid BM25 + dense search, MRR-optmized training and such.

We concluded to use Dense similarity search due to the reason that its the leanest way to turn every incoming call or SMS into a "nearest-neighbour lookup" against your curated scam-script corpus. Because scam language is usually short, often paraphrased, and the whole pipeline must run in <1 s on phone-class hardware, a single-stage vector search keeps recall high while avoiding the CPU, RAM, and engineering overhead of BM25 filters or cross-encoder re-rankers.

L Full Code, Live App and Workflow

Full source code for our application and link to demo video can be found in this repository *https://github.com/anand-adwaith/AI-FraudCall-Detector*. Visit the live app at *https://frauddetectionapp.publicvm.com/*

The application offers 3 input methods: Text message, Audio uploads, and Live Audio recording. For Audio Upload and Live Recording Page, the user can select Model as well as Language of input. For the Text page, the user can select only Model. The available Models options are BERT, Gemma-fine-tuned, Gemma-RAG, Gemma-FewShot, GPT-RAG and GPT-FewShot.

Text messages are directly sent to the AI Backend. For audio uploads and live audio recordings the files are temporarily stored, which is then passed to ASR component to transcribe to English text. The translated text is sent to the AI Backend. It performs three main functions: Scam classification, Reasoning, and Follow up Questions.

After analysis, the user is presented with the results (Scam, Not Scam Suspicious) , reasoning, and Follow-up questions if any, so that the user can ask the caller to get further information, thereby facilitating more informed decision making.

This comprehensive flow enables to ingest various forms of user input, transform into a text format and leverage LLM for Scam Detection

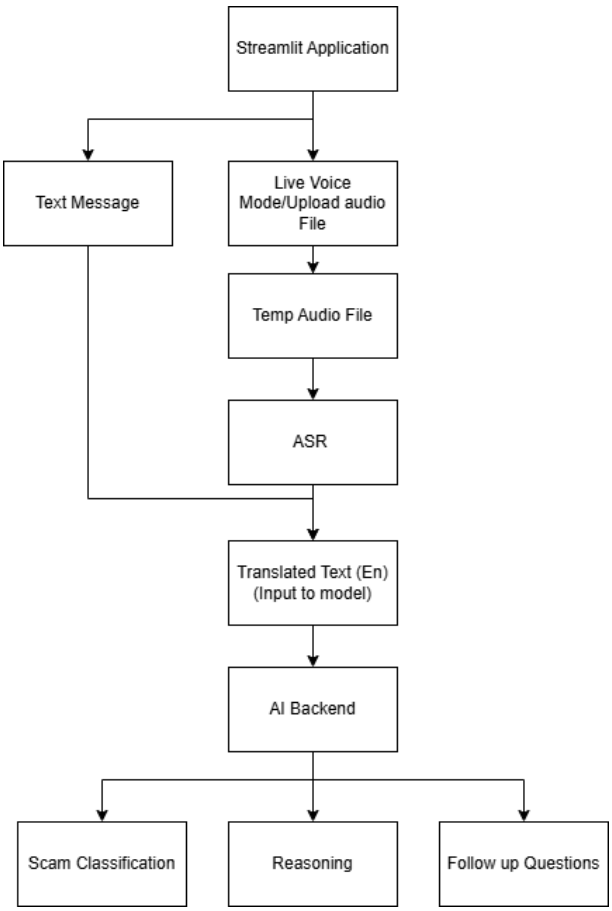

Figure 3. End to End Workflow of Application .

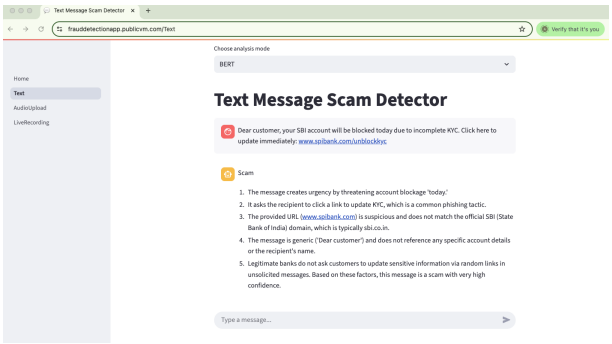

Figure 4. Text Submission Page of Application

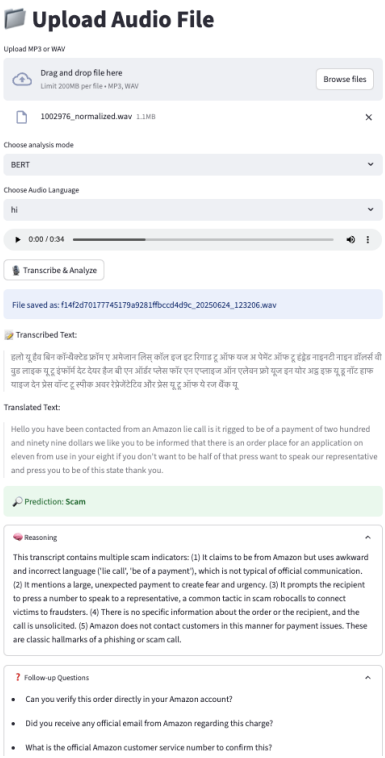

Figure 5. Audio Upload Page of Application

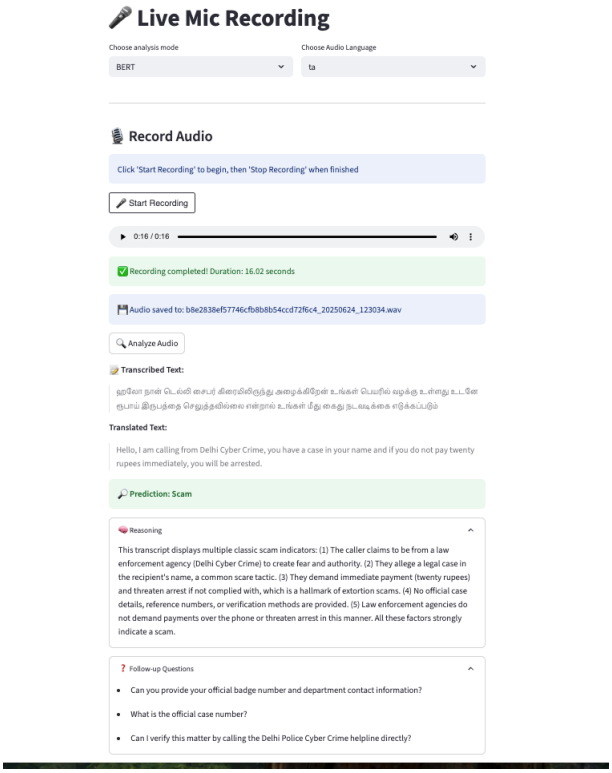

Figure 6. Audio Live Recording Page of Application