ScamGuard: Enhancing Online Safety with Scam Detection and LLM-based Explanation Insights

A. Shahnawaz, A. Zahid, D. K. Khan, M. S. Abbas, N. Hakim, and Ubaid
Lahore University of Management Sciences
{23030024, 25100247, 24100139, 20030038, 24100131, 23030027}@lums.edu.pk

Abstract

ScamGuard is a comprehensive solution for detecting online scams and providing insightful explanations. Our system integrates multiple large language models (LLMs) to identify potential scams in text or audio inputs, making it accessible to diverse users, including those who only know Urdu. By leveraging Streamlit for integration of speech-to-text module, LLMs, and a minimal interface, ScamGuard provides consolidated predictions and detailed explanations for why a text might be scam. The system's primary components include six LLMs, 6 for scam predictions and one for consoldiation of results from all LLMs, a Whisper-based ASR module, and an intuitive web-based interface. All LLMs showed similar performance on English but Gemini and GPT excelled in Urdu as well, therefore we switched between the two as consolidators. Testing showed impressive performance both in English and Arabic Script Urdu.

1 Implementation details

1.1 System Design

Our system design involved the following steps: 1) Stacking multiple LLMs for scam detection, 2) Adding a speech-to-text module to allow audio input, and 3) Integration of LLMs and ASR with Streamlit and designing the user interface.

Scam Detection and Explanation

With the goal of facilitating easier scam detection, we developed ScamGuard, a solution that processes user input to classify whether it indicates a scam or not. To make it more accessible to the Pakistani population and individuals with limited literacy, we included an audio feature where users can provide an audio description or a recording of the potential scam.

We leveraged Streamlit to seamlessly integrate our Automatic Speech Recognition(ASR), Large Language Models(LLMs), and User Interface(UI). We chose Streamlit for its rapid prototyping capabilities, smooth integration with existing machine learning pipelines, and a clean user interface. For these reasons, we were able to launch a fully functional, ML-powered web application on a local host.

The output of ScamGuard includes individual predictions from each LLM on whether the text or audio input is a scam, a consolidated probability score derived from the outputs of all LLMs, and a detailed explanation outlining the reasons why the input might be considered a scam.

Speech-to-text Module

Our LLM module requires text as input for scam classification. To handle audio input, we convert it to text as a preprocessing step using the Whisper-base model. Whisper (Radford et al., 2022), a Transformer-based encoder-decoder model, or sequence-to-sequence model, is trained on 680k hours of labeled speech data using large-scale weak supervision. The models are trained on either English-only or multilingual data and are available in five configurations of varying sizes. We chose Whisper-base (74M parameters), a multilingual model, to cater to audio input in both Urdu and English. This ensures that our application can effectively classify scams from Urdu audio inputs.

We tested the performance of whisper-base model with various English and Urdu input audios and it demonstrated remarkable generalization across diverse domains without requiring fine-tuning, which is why we opted not to fine-tune it for our application. The model's performance only deteriorated when background noise was excessive and overshadowed the primary audio. Consequently, this limitation implies that our application requires audio input with moderate noise levels for optimal performance.

LLMs Component

For performance evaluation and selection of large language models (LLMs), we consulted the literature. (Koide et al., 2024) analyzed four models: GPT-3.5-Turbo, GPT-4, Llama2-70B, and Gemini Pro, all achieving over 95% accuracy in spam detection. Similarly, (Patel et al., 2024) assessed various LLMs, including GPT-4, GPT-3.5, Llama2, and Mistral-7B, on a subset of Kaggle's fraudulent email corpus. Consistently across studies, the GPT models performed exceptionally well in identifying spam, scam, and phishing attempts. The decoder-only design allows these models to effectively comprehend natural language flow enabling the identification of inconsistencies that hint at phishing and potential fraudulent scenarios.

Using both literature and our own testing, we selected seven LLMs:

1. **Mistral 8x22B** is open-source, with 64K tokens context windo. It is a sparse Mixture-of-Experts (SMoE) model, and activates only 39 billion of its 141 billion parameters, providing exceptional cost efficiency.

2. **Llama 3-70B** is an open-source auto-regressive language model with an optimized transformer architecture, pre-trained on over 15 trillion tokens sourced from publicly available data. It is fine-tuned with instruction datasets and 10 million human-annotated examples.

3. **Phi-2** is an open-source, 2.7 billion-parameter Transformer trained with data sources from Phi-1.5, augmented by synthetic NLP texts and carefully filtered websites. Phi-2 demonstrates near state-of-the-art performance among models with fewer than 13 billion parameters, excelling in common sense, language understanding, and logical reasoning.

4. **StripedHyena-Nous-7B** is the first alternative model that can compete with the top open-source Transformers in both short- and long-context tasks. It matches or outperforms models like Llama-2 and Mistral 7B on OpenLLM leaderboard tasks, particularly excelling in long-context summarization. It employs grafting techniques to incorporate the best components of Transformers and Hyena architectures.

5. **OLMo-7B-instruct** is open-source and leverages the Allen AI's Dolma dataset, which contains a 3 trillion token open corpus for model pre-training. With four 7B variants trained to at least 2 trillion tokens each, this model provides extensive checkpoints, inference code, and evaluation metrics.

6. **Gemma-7B** is a family of lightweight, open models built on the research used for Google's Gemini models. These decoder-only, text-to-text models are adept at tasks like question answering, summarization, and reasoning. Despite their smaller size, they are versatile and efficient enough to run on laptops or local cloud infrastructure. They are trained on 6 trillion tokens from diverse text sources.

7. **GPT-3.5-Turbo** is an optimized variant of OpenAI's GPT-3 model family. It supports a wide range of tasks such as summarization, translation, and conversational understanding, while offering flexible deployment options. This model has been instrumental in providing high-quality answers across numerous domains due to its ability to capture and predict contextual language patterns effectively, as evident in literature.

8. **Gemini-1.0** is a state-of-the-art LLM developed by Google. These model is known for its versatility and efficiency in tasks such as question answering, summarization, and reasoning. Built with a decoder-only architecture and trained on a vast corpus of high-quality, diverse text, Gemini model is optimized for natural language understanding and generation. Its provide good performance across multiple languages and topics, making it suitable for a wide range of applications.

The input text is passed through each of the six LLMs, generating predictions and explanations. These outputs are then passed to GPT-3.5(or Gemini), which serves as a consolidator, merging the predictions and providing with a explanation for why a text might be a scam.

Initially, GPT-3.5 was used as the final LLM in our system, consolidating the outputs of all other models. However, we ran out of tokens during testing and evaluation, and switched to Gemini to perform this role. Figure 1 illustrates the current system design, which excludes GPT-3.5, although it was originally included and demonstrated superior performance. In the final system design, we have 6 predictor LLMs and one consolidator LLM.

2 Deployment

The app repository is hosted on GitHub, containing the Streamlit app. To run the app, first pull the repository. Ensure that Streamlit and Whisper are installed before executing this command. You can install Streamlit with:

```
pip install streamlit
```

For Whisper, use:

```
pip install git+https://github.com/
    openai/whisper.git
```

To install everything at once, you can run requiremnets.txt.

You'll also need to run the following command in windows powershell to install chocolatey that will then, set up the ffmpeg for whisper:

```
Set-ExecutionPolicy Bypass -Scope
    Process -Force;
[System.Net.ServicePointManager]::
    SecurityProtocol =
[System.Net.ServicePointManager]::
    SecurityProtocol -bor 3072;
iex ((New-Object System.Net.WebClient).
    DownloadString('https://chocolatey.
    org/install.ps1'))
```

Next, install ffmpeg via Chocolatey:

```
choco install ffmpeg
```

This setup will ensure you have everything required to run the app. Now, navigate to the appropriate directory, and use the command:

```
streamlit run ScamGuard-main/View/app.py
```

This command will start app for you in the browser.

The app is hosted locally, allowing the user to interact with it through a web browser. After following the setup instructions and running the app, you can access it by opening a web browser and navigating to the appropriate local address, typically http://localhost:8501. The browser will provide interface for interacting with the Streamlit application. The idea behind using browser as interface was to provide users with a simple and uncomplicated way to interact with the app. Streamlit allowed for rapid prototyping capabilities and smooth integration with existing machine learning pipelines. We plan to dockerize our Streamlit app and host it on Google Cloud(Thanks to their $300 free credit offer).

3 Experiments and Iterations

To improve , we employed a systematic, iterative approach. First, we tested LLM individually using various prompts, to get insights into their respective strengths and weaknesses, particularly with Urdu language. By analyzing their responses to different prompts, we identified areas where performance lacked and where they excelled. This detailed analysis allowed us to refine a specialized prompt template tailored specifically for the consolidator LLM(GPT) which is shown in Figure 2. This template became the standard by which the consolidator LLM would interpret and analyze the text input and outputs of individual LLMs. Our refinements include clarifying instructions and emphasizing key aspects such as identifying impersonation and social engineering tactics. We later switched to using Gemini for the consolidation process. In doing so, we refined the prompt initially created for GPT by incorporating additional details. This prompt not only includes the predicted probability of each input being a scam, but also gathers and presents the individual predictions and total probability of scam. By providing this level of granular information, the updated prompt enables Gemini to assess and interpret the results from all contributing models, producing a more detailed explanation for the classification. This approach provides more comprehensive insights to the user.

Figures 4 and 6 demonstrate sample outputs generated by both GPT and Gemini as consolidators. The results reveal that GPT consistently produced more cohesive responses when dealing with Urdu scam messages. This cohesiveness ensured that scam indicators were accurately identified and explained in a structured manner, enabling comprehensive evaluations.

4 Reflections

What worked well? Initially, we intended to use a separate machine learning model to classify text and then have an LLM provide explanations, as outlined in our project proposal. However, after

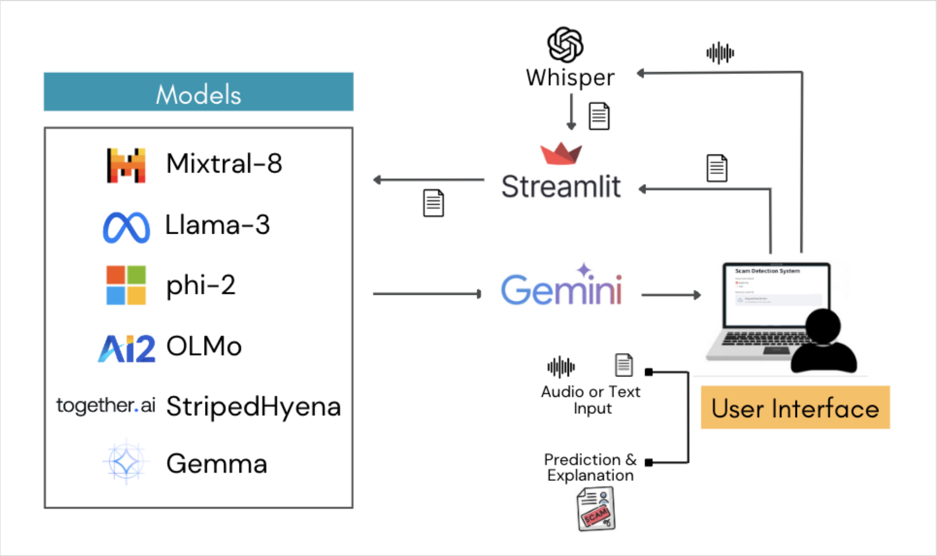


Figure 1: Overview of ScamGuard's System Design, application's component interactions and data flow

experimenting with several LLMs, we found that their scam detection surpassed that of our classifier. As a result, we used LLMs for both classification and explanation. Among the options tested, GPT and Gemini worked best for Urdu, but GPT delivered the strongest overall performance.

What didn't work well? Our GPT credits ran out on the final day, so we had to use Gemini for consolidation. While it performed adequately compared to GPT, it didn't fully match up. Additionally, many LLMs struggled to identify Urdu scams, likely due to limited support for the language. We also attempted to dockerize the Streamlit app and host it on Google Cloud, but couldn't complete it within the project timeline.

What would be the next step to improving your system? A valuable next step would be to gather a comprehensive dataset of scams over time and fine-tune a model using this data. We aim to deploy it in the cloud to see if it can effectively help the general population recognize scam tactics and educate themselves on common indicators.

How could you have remedied what went wrong? Overall, nothing major went wrong—just minor issues like prompt misfires and some trial-and-error with API integration. The primary challenge was running out of tokens, which could have been mitigated by not being broke.

5 Demonstration

Figures 7 shows the main landing page of the application and outputs for text and audio. 6 shows the output of system with the final prompt.

6 Contribution

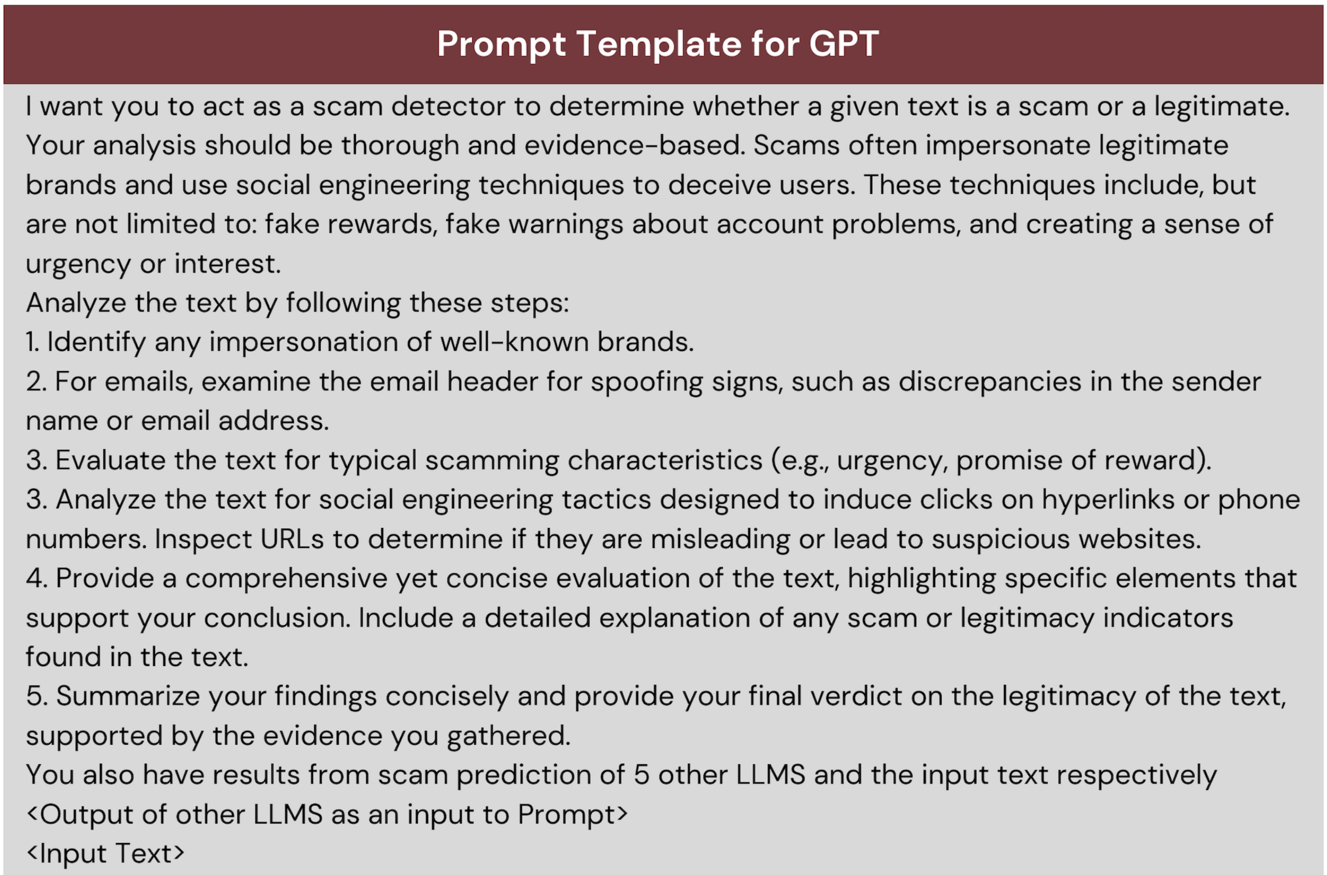
All members worked on conceptual system design. Areeb and Sameer worked on LLMs integration in Streamlit. Ubaid worked on user interface. Naveed handled the maintenance of Github repository, integrated Whisper module with streamlit and assissted Ubaid in UI integration. Amna authored the report, refined the GPT prompts and assisted Areeb with Gemini prompts.

References

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. *Robust Speech Recognition via Large-Scale Weak Supervision*. arXiv preprint arXiv:2212.04356.

Takashi Koide, et al. 2024. *ChatSpamDetector: Leveraging Large Language Models for Effective Phishing Email Detection*. arXiv preprint arXiv:2402.18093.

Het Patel, Umair Rehman, and Farkhund Iqbal. 2024. *Large Language Models Spot Phishing Emails with Surprising Accuracy: A Comparative Analysis of Performance*. arXiv preprint arXiv:2404.15485.

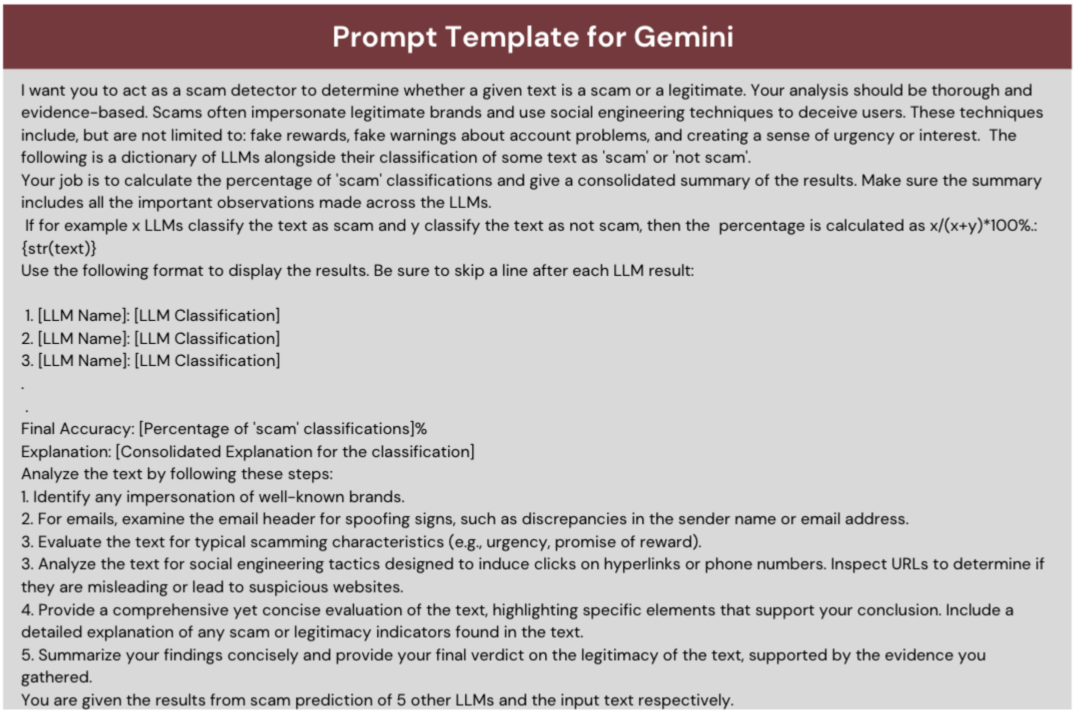


Prompt Template for GPT

I want you to act as a scam detector to determine whether a given text is a scam or a legitimate. Your analysis should be thorough and evidence-based. Scams often impersonate legitimate brands and use social engineering techniques to deceive users. These techniques include, but are not limited to: fake rewards, fake warnings about account problems, and creating a sense of urgency or interest.
Analyze the text by following these steps:
1. Identify any impersonation of well-known brands.
2. For emails, examine the email header for spoofing signs, such as discrepancies in the sender name or email address.
3. Evaluate the text for typical scamming characteristics (e.g., urgency, promise of reward).
3. Analyze the text for social engineering tactics designed to induce clicks on hyperlinks or phone numbers. Inspect URLs to determine if they are misleading or lead to suspicious websites.
4. Provide a comprehensive yet concise evaluation of the text, highlighting specific elements that support your conclusion. Include a detailed explanation of any scam or legitimacy indicators found in the text.
5. Summarize your findings concisely and provide your final verdict on the legitimacy of the text, supported by the evidence you gathered.

You also have results from scam prediction of 5 other LLMS and the input text respectively
<Output of other LLMS as an input to Prompt>
<Input Text>

Figure 2: Prompt Template for GPT as consolidator LLM



Prompt Template for Gemini

I want you to act as a scam detector to determine whether a given text is a scam or a legitimate. Your analysis should be thorough and evidence-based. Scams often impersonate legitimate brands and use social engineering techniques to deceive users. These techniques include, but are not limited to: fake rewards, fake warnings about account problems, and creating a sense of urgency or interest. The following is a dictionary of LLMs alongside their classification of some text as 'scam' or 'not scam'.
Your job is to calculate the percentage of 'scam' classifications and give a consolidated summary of the results. Make sure the summary includes all the important observations made across the LLMs.
If for example x LLMs classify the text as scam and y classify the text as not scam, then the percentage is calculated as x/(x+y)*100%.:
{str(text)}
Use the following format to display the results. Be sure to skip a line after each LLM result:

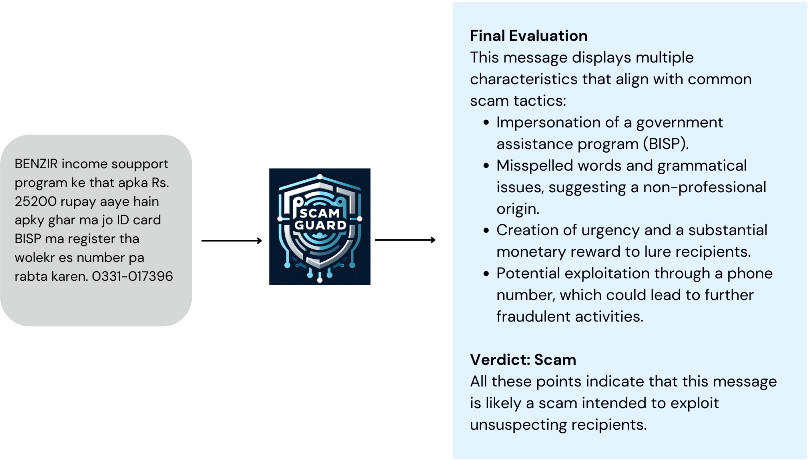
1. [LLM Name]: [LLM Classification]
2. [LLM Name]: [LLM Classification]
3. [LLM Name]: [LLM Classification]

.
.

Final Accuracy: [Percentage of 'scam' classifications]%
Explanation: [Consolidated Explanation for the classification]
Analyze the text by following these steps:
1. Identify any impersonation of well-known brands.
2. For emails, examine the email header for spoofing signs, such as discrepancies in the sender name or email address.
3. Evaluate the text for typical scamming characteristics (e.g., urgency, promise of reward).
3. Analyze the text for social engineering tactics designed to induce clicks on hyperlinks or phone numbers. Inspect URLs to determine if they are misleading or lead to suspicious websites.
4. Provide a comprehensive yet concise evaluation of the text, highlighting specific elements that support your conclusion. Include a detailed explanation of any scam or legitimacy indicators found in the text.
5. Summarize your findings concisely and provide your final verdict on the legitimacy of the text, supported by the evidence you gathered.

You are given the results from scam prediction of 5 other LLMs and the input text respectively.

Figure 3: Prompt Template for Gemini as consolidator LLM



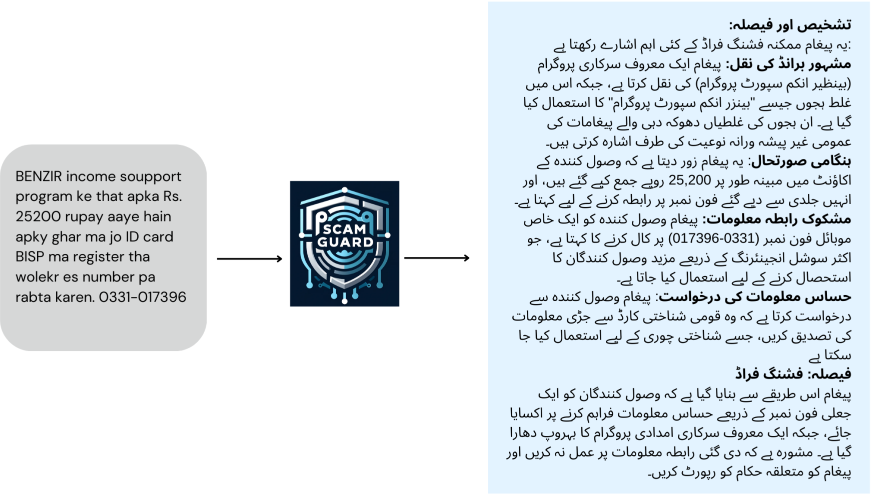
BENZIR income soupport program ke that apka Rs. 25200 rupay aaye hain apky ghar ma jo ID card BISP ma register tha wolekr es number pa rabta karen. 0331-017396

Final Evaluation
This message displays multiple characteristics that align with common scam tactics:

- Impersonation of a government assistance program (BISP).
- Misspelled words and grammatical issues, suggesting a non-professional origin.
- Creation of urgency and a substantial monetary reward to lure recipients.
- Potential exploitation through a phone number, which could lead to further fraudulent activities.

Verdict: Scam
All these points indicate that this message is likely a scam intended to exploit unsuspecting recipients.

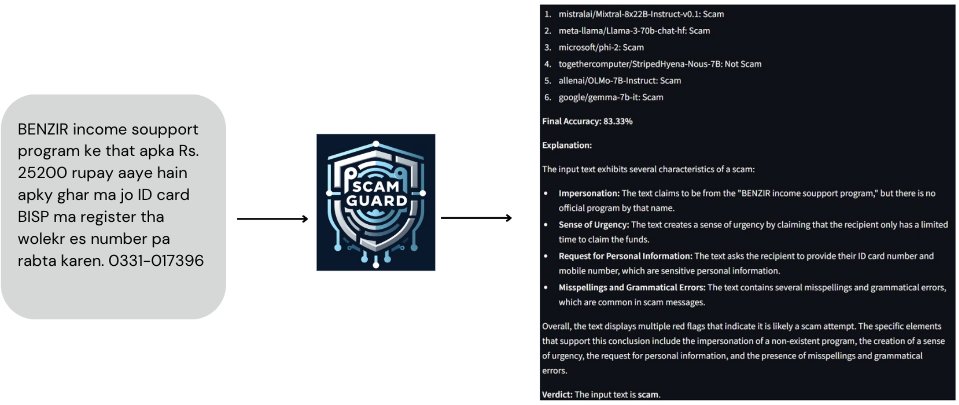
Figure 4: Scam detection using GPT consolidator LLM with English Explanation



BENZIR income soupport program ke that apka Rs. 25200 rupay aaye hain apky ghar ma jo ID card BISP ma register tha wolekr es number pa rabta karen. 0331-017396

تشخیص اور فیصلہ:
:یہ پیغام ممکنہ فشنگ فراڈ کے کئی اہم اشارے رکھتا ہے
مشہور برانڈ کی نقل: پیغام ایک معروف سرکاری پروگرام (بینظیر انکم سپورٹ پروگرام) کی نقل کرتا ہے، جبکہ اس میں غلط ہجوں جیسے "بینزر انکم سپورٹ پروگرام" کا استعمال کیا گیا ہے۔ ان ہجوں کی غلطیاں دھوکہ دہی والے پیغامات کی عمومی غیر پیشہ ورانہ نوعیت کی طرف اشارہ کرتی ہیں۔
ہنگامی صورتحال: یہ پیغام زور دیتا ہے کہ وصول کنندہ کے اکاؤنٹ میں مبینہ طور پر 25,200 روپے جمع کیے گئے ہیں، اور انہیں جلدی سے دیے گئے فون نمبر پر رابطہ کرنے کے لیے کہتا ہے۔
مشکوک رابطہ معلومات: پیغام وصول کنندہ کو ایک خاص موبائل فون نمبر (0331-017396) پر کال کرنے کا کہتا ہے، جو اکثر سوشل انجینئرنگ کے ذریعے مزید وصول کنندگان کا استحصال کرنے کے لیے استعمال کیا جاتا ہے۔
حساس معلومات کی درخواست: پیغام وصول کنندہ سے درخواست کرتا ہے کہ وہ قومی شناختی کارڈ سے جڑی معلومات کی تصدیق کریں، جسے شناختی چوری کے لیے استعمال کیا جا سکتا ہے
فیصلہ: فشنگ فراڈ
پیغام اس طریقے سے بنایا گیا ہے کہ وصول کنندگان کو ایک جعلی فون نمبر کے ذریعے حساس معلومات فراہم کرنے پر اکسایا جائے، جبکہ ایک معروف سرکاری امدادی پروگرام کا بہروپ دھارا گیا ہے۔ مشورہ ہے کہ دی گئی رابطہ معلومات پر عمل نہ کریں اور پیغام کو متعلقہ حکام کو رپورٹ کریں۔

Figure 5: Scam detection using GPT consolidator LLM with Urdu Explanation


BENZIR income soupport program ke that apka Rs. 25200 rupay aaye hain apky ghar ma jo ID card BISP ma register tha wolekr es number pa rabta karen. 0331-017396

1. mistralai/Mixtral-8x22B-Instruct-v0.1: Scam
2. meta-llama/Llama-3-70b-chat-hf: Scam
3. microsoft/phi-2: Scam
4. togethercomputer/StripedHyena-Nous-7B: Not Scam
5. allenai/OLMo-7B-Instruct: Scam
6. google/gemma-7b-it: Scam

Final Accuracy: 83.33%

Explanation:

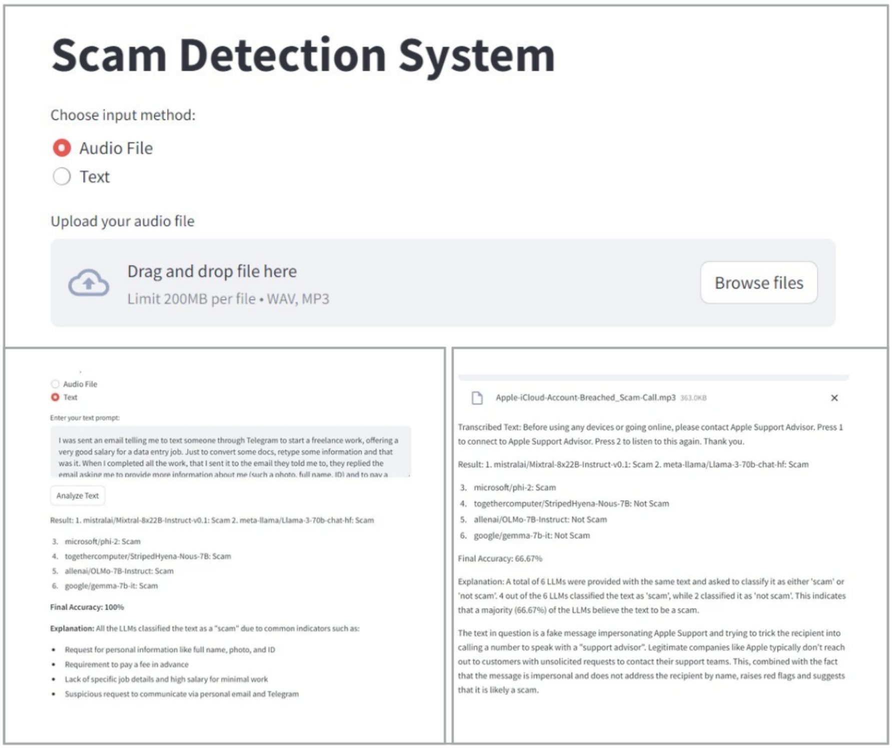
The input text exhibits several characteristics of a scam:

- **Impersonation:** The text claims to be from the "BENZIR income soupport program," but there is no official program by that name.
- **Sense of Urgency:** The text creates a sense of urgency by claiming that the recipient only has a limited time to claim the funds.
- **Request for Personal Information:** The text asks the recipient to provide their ID card number and mobile number, which are sensitive personal information.
- **Misspellings and Grammatical Errors:** The text contains several misspellings and grammatical errors, which are common in scam messages.

Overall, the text displays multiple red flags that indicate it is likely a scam attempt. The specific elements that support this conclusion include the impersonation of a non-existent program, the creation of a sense of urgency, the request for personal information, and the presence of misspellings and grammatical errors.

Verdict: The input text is **scam.**

Figure 6: Scam detection using Gemini consolidator LLM



I was sent an email telling me to text someone through Telegram to start a freelance work, offering a very good salary for a data entry job. Just to convert some docs, retype some information and that was it. When I completed all the work, that I sent it to the email they told me to, they replied the email asking me to provide more information about me (such a photo, full name, ID) and to pay a

Result: 1. mistralai/Mixtral-8x22B-Instruct-v0.1: Scam 2. meta-llama/Llama-3-70b-chat-hf: Scam

3. microsoft/phi-2: Scam
4. togethercomputer/StripedHyena-Nous-7B: Scam
5. allenai/OLMo-7B-Instruct: Scam
6. google/gemma-7b-it: Scam

Final Accuracy: 100%

Explanation: All the LLMs classified the text as a "scam" due to common indicators such as:

- Request for personal information like full name, photo, and ID
- Requirement to pay a fee in advance
- Lack of specific job details and high salary for minimal work
- Suspicious request to communicate via personal email and Telegram

Transcribed Text: Before using any devices or going online, please contact Apple Support Advisor. Press 1 to connect to Apple Support Advisor. Press 2 to listen to this again. Thank you.

Result: 1. mistralai/Mixtral-8x22B-Instruct-v0.1: Scam 2. meta-llama/Llama-3-70b-chat-hf: Scam

3. microsoft/phi-2: Scam
4. togethercomputer/StripedHyena-Nous-7B: Not Scam
5. allenai/OLMo-7B-Instruct: Not Scam
6. google/gemma-7b-it: Not Scam

Final Accuracy: 66.67%

Explanation: A total of 6 LLMs were provided with the same text and asked to classify it as either 'scam' or 'not scam'. 4 out of the 6 LLMs classified the text as 'scam', while 2 classified it as 'not scam'. This indicates that a majority (66.67%) of the LLMs believe the text to be a scam.

The text in question is a fake message impersonating Apple Support and trying to trick the recipient into calling a number to speak with a "support advisor". Legitimate companies like Apple typically don't reach out to customers with unsolicited requests to contact their support teams. This, combined with the fact that the message is impersonal and does not address the recipient by name, raises red flags and suggests that it is likely a scam.

Figure 7: Application Demonstartion: Top Image: Main landing page, Bottom Left and right: Scam detection using text and audio input respectively