User Manual

1. Python application **3**
1.1. Overview **3**
1.2. Requirements **3**
1.3. Usage **3**
1.4. Viewing Pre-made Results **4**
1.5. Running Individual Algorithm Modules **4**
2. Telegram bot **5**
2.1. Overview **5**
2.2. Get Started **5**
2.3. Features **5**
3. Android Mobile application **9**
3.1. Application Overview **9**
3.2. System summary **9**
  3.2.1. Hardware and Software Requirements 9
  3.2.2. Installation 9
3.3. Features **11**
  3.3.1. Home page 11
  3.3.2. Enter new message 15
  3.3.3. View previous messages 16

1. Python application

1.1. Overview

The compare_algos.py *script is designed to compare the efficiency and accuracy of various modules for vishing detection. It evaluates different algorithms available in the "modules" folder, including cosine similarity, gensim, kmp, nltk, sklearn, spacy, and trie modules. By running this script, you can get insights into how each algorithm performs across 300+ test cases in terms of time efficiency and accuracy.*

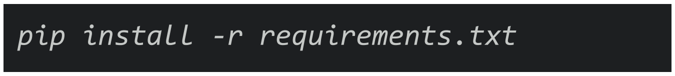
1.2. Requirements

Before running the compare_algos.py script, ensure that you have the required libraries installed on your system. You can do this by following these steps:

Open your terminal or command prompt.

Navigate to the "01-python-application" directory, which contains the compare_algos.py script, the requirements.txt file, the compare_algos_results.txt file, and the "detection_modules" folder.

Install the required libraries by running the following command:

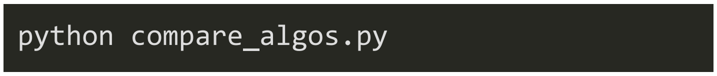


```
pip install -r requirements.txt
```

This will install all the necessary libraries listed in the requirements.txt file.

1.3. Usage

1. Open your terminal or command prompt.
2. Navigate to the "01-python-application" directory that contains the relevant files.
3. To compare all algorithms with 300 test cases, execute the following command:



```
python compare_algos.py
```

Wait for the script to complete the comparison process. Please note that running 300+ test cases for all algorithms might take some time.

1.4. Viewing Pre-made Results

To expedite the viewing of the final result after running the script, a pre-made text file named **compare_algos_results.txt** is provided with the comparison results. This file will show the efficiency and accuracy of each algorithm for the 300 test cases. You can access this file after running the script. The file will be generated in the "01-python-application" directory.

1.5. Running Individual Algorithm Modules

If you want to evaluate only one specific algorithm, you can run the respective algorithm module directly from the "detection_modules" folder. Each module is designed to handle 300 test cases for the specific algorithm it represents.

1. Open your terminal or command prompt.

2. Navigate to the "01-python-application" directory that contains the "detection_modules" folder.

3. Choose the algorithm module you want to evaluate:
    a. trie_vishing_detection.py
    b. kmp_vishing_detection.py
    c. cosineSim_vishing_detection.py
    d. gensim_vishing_detection.py
    e. nltk_vishing_detection.py
    f. sklearn_vishing_detection.py
    g. spacy_vishing_detection.py.

4. Execute the following command:

```
python detection_modules/<<algorithm_module>>
```

   Replace "<<algorithm_module>>" with the filename of the specific algorithm module you want to run.

5. Wait for the script to complete the evaluation process for that algorithm.

2. Telegram bot

2.1. Overview

The Vishing Detector Bot in telegram is a comprehensive tool designed to help users identify and prevent vishing attacks. Its main features allow users to:

1. Test and analyze text messages or audios for potential vishing content.
2. Change the vishing detection module in use to test different detection methods.
3. Compare the performance of different vishing detection modules, either in real-time or by accessing pre-prepared comparison results.
4. Participate in a vishing detection quiz to see likelihood of vishing base on the answers to the questions.

This bot employs advanced algorithms and user-friendly commands to ensure online safety is easily accessible. More details about its specific functions and usage will be explained in subsequent sections.

2.2. Get Started

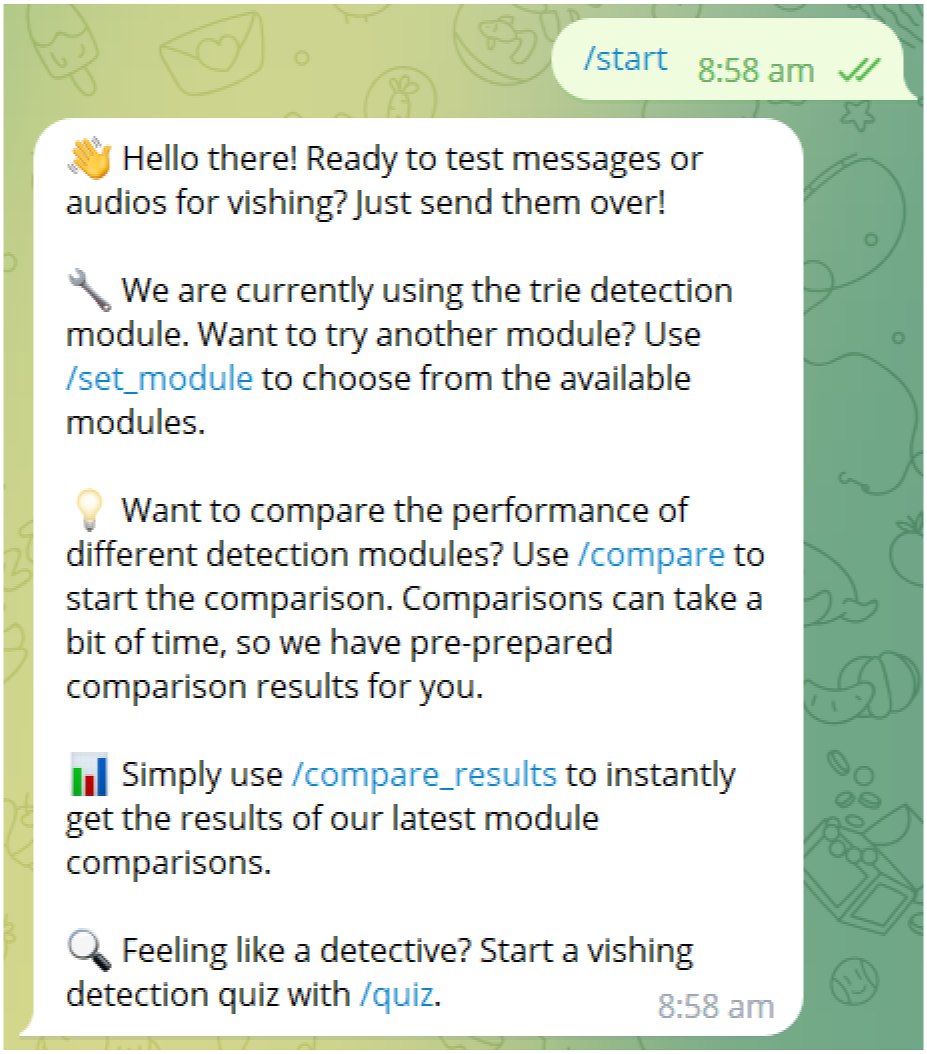
Just follow these simple steps:

1. Users need to have the Telegram app on their smartphone.
2. They can search for "Vishing Detector Bot" in the Telegram app's search bar or use the link: https://t.me/VishingDetectorBot.
3. Send the "/start" command to the bot to get instructions and see available commands.

2.3. Features

2.4.

1. **Start the bot:** Use the /start command to start the bot. The bot will greet you and explain its functionalities.



/start 8:58 am

👋 Hello there! Ready to test messages or audios for vishing? Just send them over!

🔧 We are currently using the trie detection module. Want to try another module? Use /set_module to choose from the available modules.

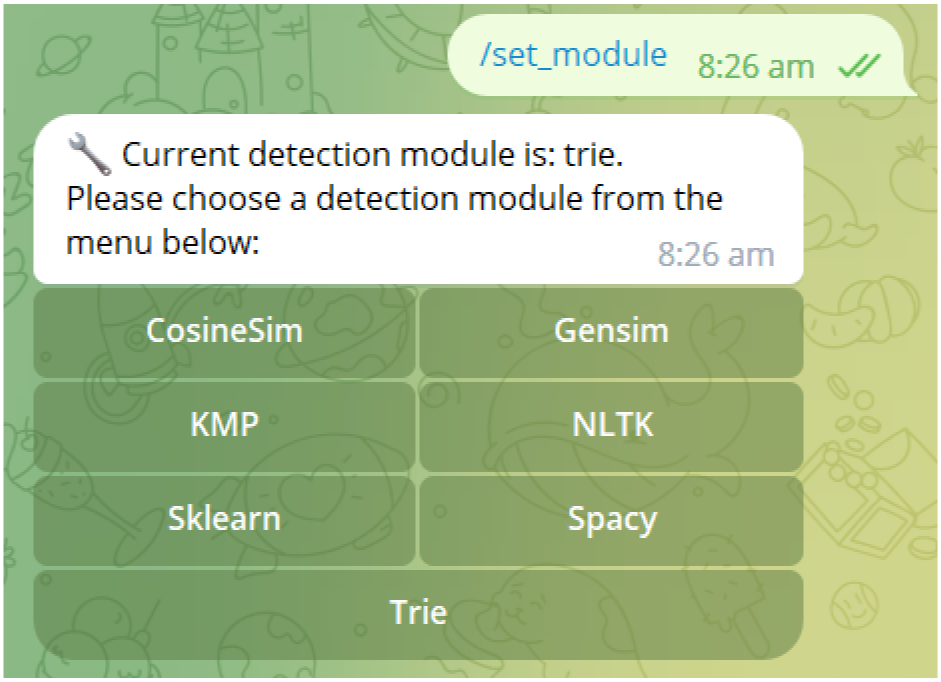
💡 Want to compare the performance of different detection modules? Use /compare to start the comparison. Comparisons can take a bit of time, so we have pre-prepared comparison results for you.

📊 Simply use /compare_results to instantly get the results of our latest module comparisons.

🔍 Feeling like a detective? Start a vishing detection quiz with /quiz.

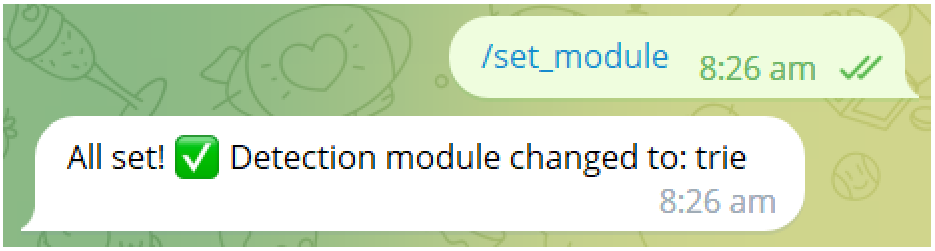
8:58 am

2. **Set detection module:** Use the /set_module command to select a detection module from the given list. The bot will then use this module to detect phishing in your messages and audios.



/set_module 8:26 am

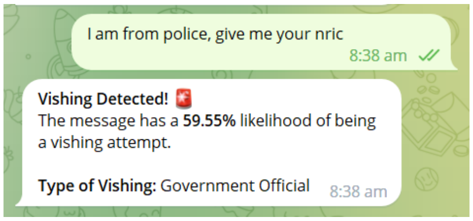
🔧 Current detection module is: trie.
Please choose a detection module from the menu below: 8:26 am



/set_module 8:26 am

All set! ✅ Detection module changed to: trie 8:26 am

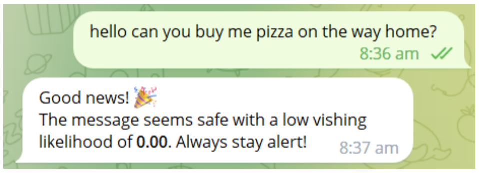
3. **Send a text message or audio message:** Just send a text or audio message to the bot as you would to any other Telegram user. The bot will analyze your message and reply with whether the message is likely to be a phishing attempt or not.



I am from police, give me your nric 8:38 am

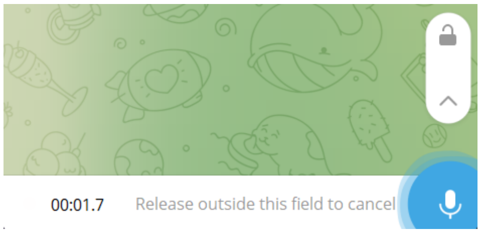
Vishing Detected! 🚨
The message has a **59.55%** likelihood of being a vishing attempt.

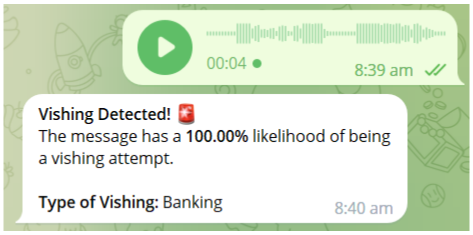
Type of Vishing: Government Official 8:38 am



hello can you buy me pizza on the way home? 8:36 am

Good news! 🎉
The message seems safe with a low vishing likelihood of **0.00**. Always stay alert! 8:37 am

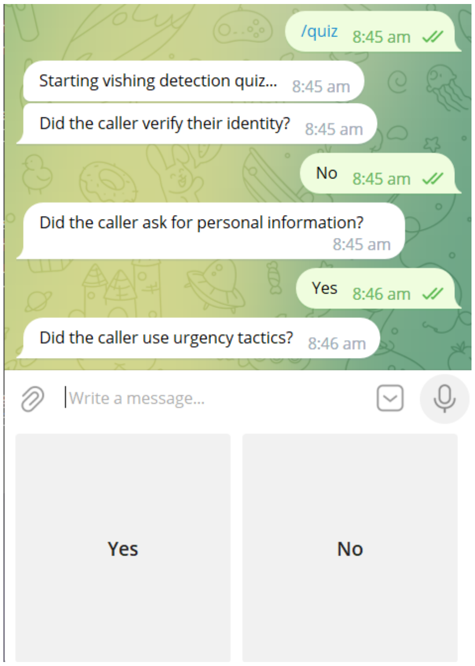




Vishing Detected! 🚨
The message has a **100.00%** likelihood of being a vishing attempt.

Type of Vishing: Banking 8:40 am

4. **Take a quiz:** Use the /quiz command to start a vishing detection quiz. The bot will ask you a series of questions to test your knowledge about phishing.



/quiz 8:45 am

Starting vishing detection quiz... 8:45 am

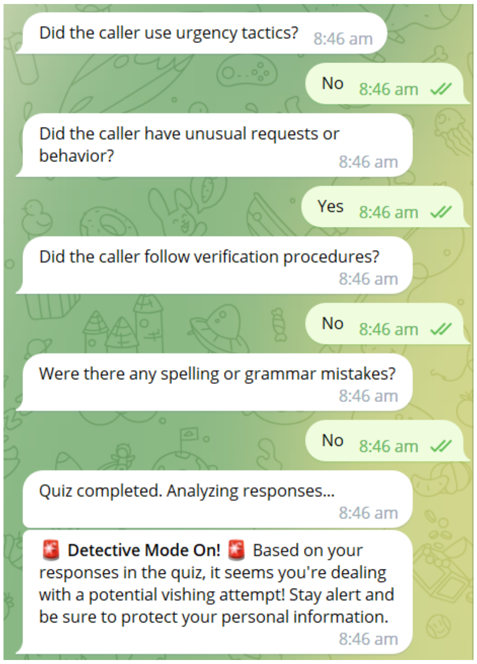
Did the caller verify their identity? 8:45 am

No 8:45 am

Did the caller ask for personal information? 8:45 am

Yes 8:46 am

Did the caller use urgency tactics? 8:46 am



Did the caller use urgency tactics? 8:46 am

No 8:46 am

Did the caller have unusual requests or behavior? 8:46 am

Yes 8:46 am

Did the caller follow verification procedures? 8:46 am

No 8:46 am

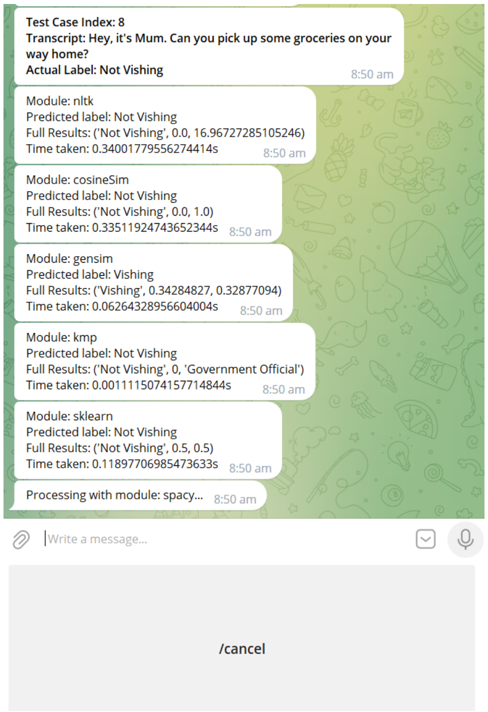
Were there any spelling or grammar mistakes? 8:46 am

No 8:46 am

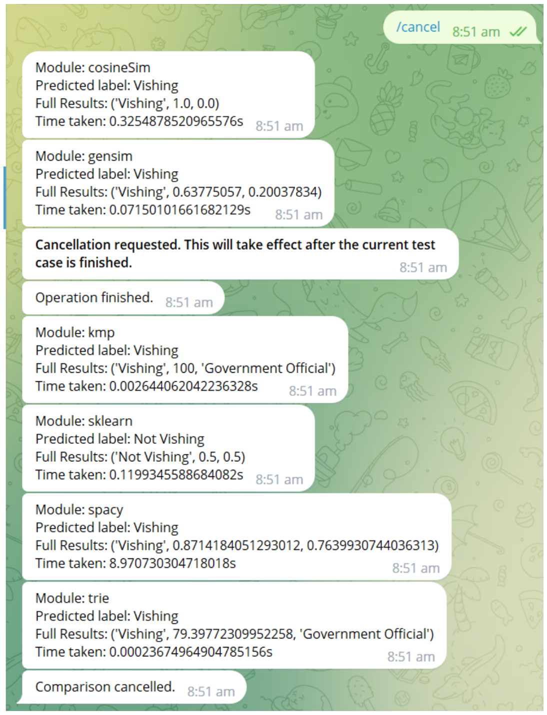
Quiz completed. Analyzing responses... 8:46 am

🚨 **Detective Mode On!** 🚨 Based on your responses in the quiz, it seems you're dealing with a potential vishing attempt! Stay alert and be sure to protect your personal information. 8:46 am

5. **Compare performance of different modules:** Use the /compare command to start a comparison of the performance of different detection modules. The bot will process a pre-defined dataset of 300+ test cases with each module and provide a comparison of their performance. The comparison can be cancelled at any time using the /cancel command. When completed it will output the conclusion with the overall result from all modules.

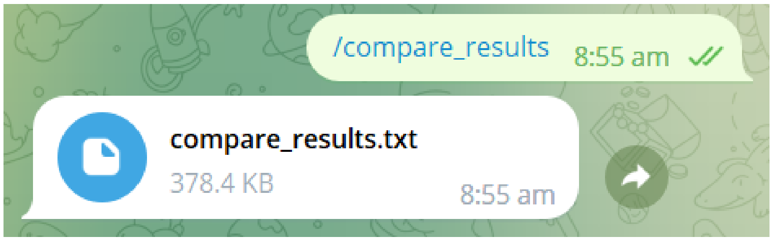


```
Test Case Index: 8
Transcript: Hey, it's Mum. Can you pick up some groceries on your way home?
Actual Label: Not Vishing                                        8:50 am

Module: nltk
Predicted label: Not Vishing
Full Results: ('Not Vishing', 0.0, 16.96727285105246)
Time taken: 0.34001779556274414s                                 8:50 am

Module: cosineSim
Predicted label: Not Vishing
Full Results: ('Not Vishing', 0.0, 1.0)
Time taken: 0.33511924743652344s                                 8:50 am

Module: gensim
Predicted label: Vishing
Full Results: ('Vishing', 0.34284827, 0.32877094)
Time taken: 0.06264328956604004s                                 8:50 am

Module: kmp
Predicted label: Not Vishing
Full Results: ('Not Vishing', 0, 'Government Official')
Time taken: 0.0011115074157714844s                               8:50 am

Module: sklearn
Predicted label: Not Vishing
Full Results: ('Not Vishing', 0.5, 0.5)
Time taken: 0.11897706985473633s                                 8:50 am

Processing with module: spacy...                                 8:50 am
```



```
/cancel 8:51 am

Module: cosineSim
Predicted label: Vishing
Full Results: ('Vishing', 1.0, 0.0)
Time taken: 0.3254878520965576s                                  8:51 am

Module: gensim
Predicted label: Vishing
Full Results: ('Vishing', 0.63775057, 0.20037834)
Time taken: 0.07150101661682129s                                 8:51 am

Cancellation requested. This will take effect after the current test
case is finished.                                                8:51 am

Operation finished.                                              8:51 am

Module: kmp
Predicted label: Vishing
Full Results: ('Vishing', 100, 'Government Official')
Time taken: 0.002644062042236328s                                8:51 am

Module: sklearn
Predicted label: Not Vishing
Full Results: ('Not Vishing', 0.5, 0.5)
Time taken: 0.1199345588684082s                                  8:51 am

Module: spacy
Predicted label: Vishing
Full Results: ('Vishing', 0.8714184051293012, 0.7639930744036313)
Time taken: 8.970730304718018s                                   8:51 am

Module: trie
Predicted label: Vishing
Full Results: ('Vishing', 79.39772309952258, 'Government Official')
Time taken: 0.00023674964904785156s                              8:51 am

Comparison cancelled.                                            8:51 am
```

6. **Check comparison results:** Use the /compare_results command to get the results of the latest module comparison executed by the bot. This is so the user does not have to wait for all 300+ test cases to be done to see the result when doing /compare.



/compare_results 8:55 am

compare_results.txt
378.4 KB 8:55 am

3. Android Mobile application

3.1. Application Overview

The vishing detection chatbot android mobile application provides a convenient way for users to check for vishing detection through entering text into the chatbot. Users are also able to make use of the google speech to text feature implemented instead of entering manually.

3.2. System summary

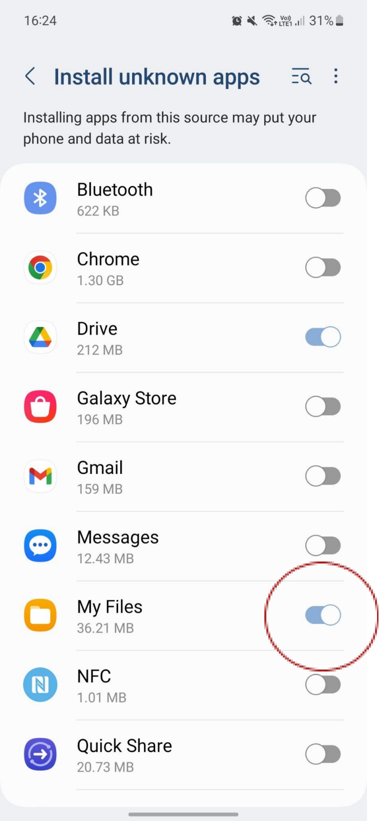
3.2.1. Hardware and Software Requirements

The following application can only be installed in mobile phones with Android operating system (OS) with a minimum requirement of android version 7.0 Nougat (API level 24).

3.2.2. Installation

The following application can be installed with the apk file (Vishing Detection Chatbot.apk) provided and upon opening the file, it will install the application to the android phone.

However, as the apk file is currently not in the app store, users may be required to ensure that they have enabled the setting to install unknown apps. They can then go to the location of the apk file and click on it to begin installation.



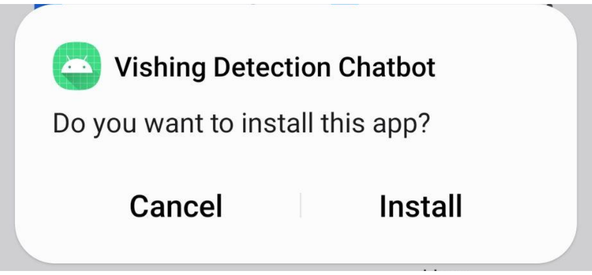
16:24 31%

Install unknown apps

Installing apps from this source may put your phone and data at risk.

Bluetooth	622 KB	○
Chrome	1.30 GB	○
Drive	212 MB	●
Galaxy Store	196 MB	○
Gmail	159 MB	○
Messages	12.43 MB	○
My Files	36.21 MB	●
NFC	1.01 MB	○
Quick Share	20.73 MB	○

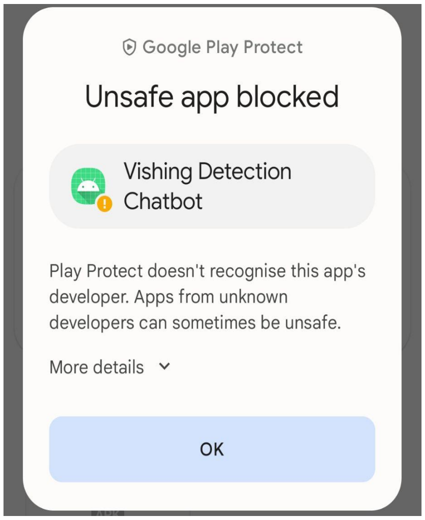
After clicking on the apk file, the following pop up will appear. Click on install.



Vishing Detection Chatbot

Do you want to install this app?

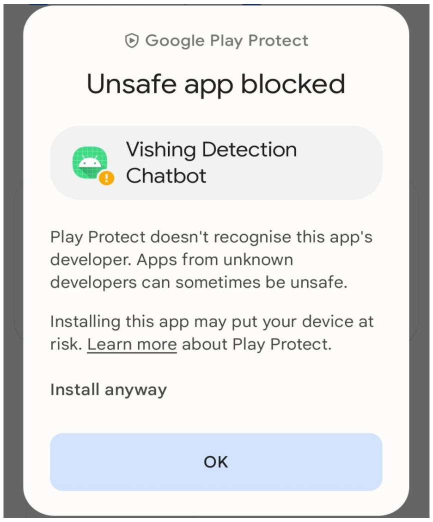
After clicking on the install option, a pop up from google play protect may appear. To proceed, click on more details for the install anyway option to appear. Click install anyway and the application should start installing. Note that if the user were to click on the ok option instead of install anyway, the application will not install.


Google Play Protect

Unsafe app blocked

Vishing Detection Chatbot

Play Protect doesn't recognise this app's developer. Apps from unknown developers can sometimes be unsafe.


Google Play Protect

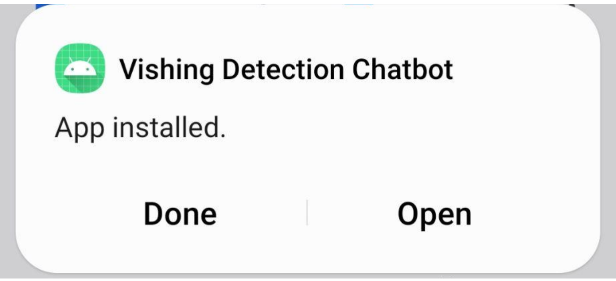
Unsafe app blocked

Vishing Detection Chatbot

Play Protect doesn't recognise this app's developer. Apps from unknown developers can sometimes be unsafe.

Installing this app may put your device at risk. Learn more about Play Protect.

Once the application is successfully installed, users will see this pop up where they can choose to open the application or click done to close the pop up.

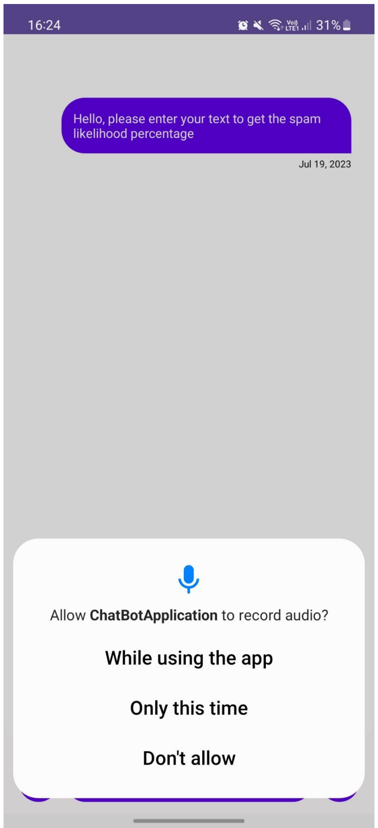

Vishing Detection Chatbot

App installed.

3.3. Features

3.3.1. Home page

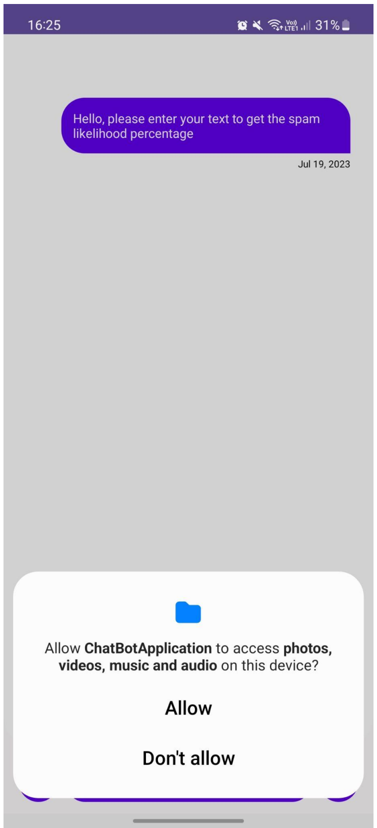
When it is the first time a user has opened the application, it will request the user to grant permissions upon startup before they can use the application.



Hello, please enter your text to get the spam likelihood percentage

Jul 19, 2023

Allow **ChatBotApplication** to record audio?

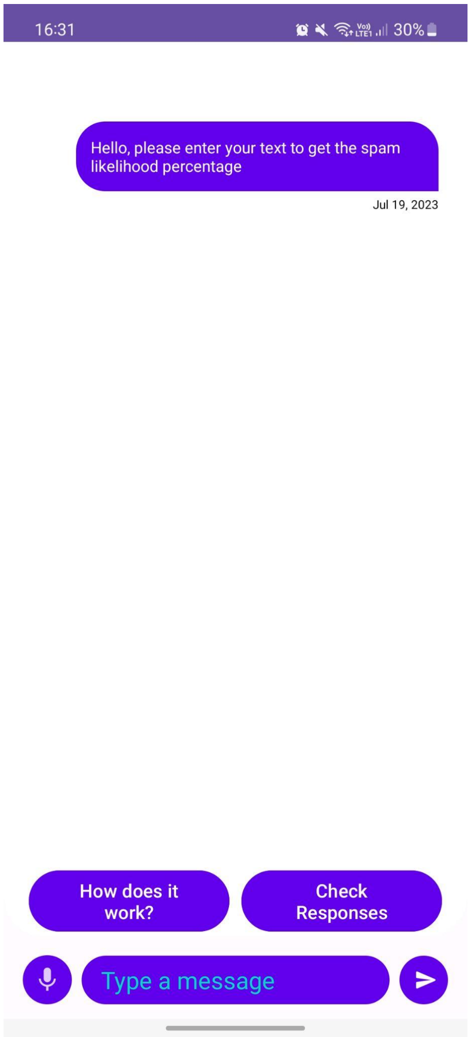


Hello, please enter your text to get the spam likelihood percentage

Jul 19, 2023

Allow **ChatBotApplication** to access **photos, videos, music and audio** on this device?

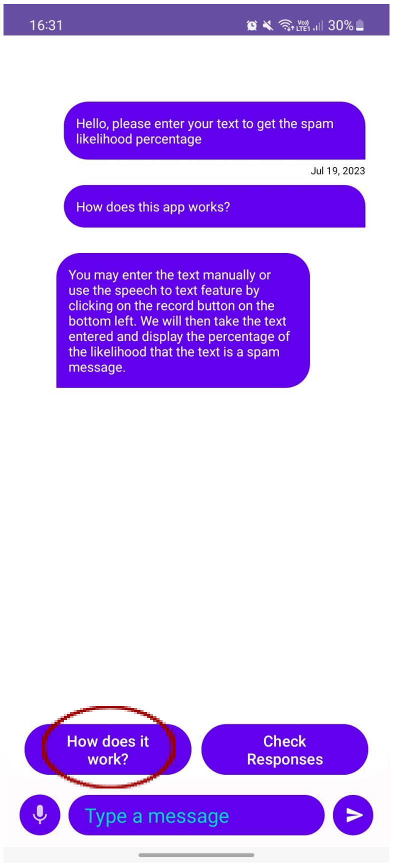
After granting permissions, the user will see the following screen.



Hello, please enter your text to get the spam likelihood percentage

Jul 19, 2023

Upon clicking on the ‘How does it work?’ button, the user will be able to see a paragraph of text explaining how the application works and what the user should do.



Hello, please enter your text to get the spam likelihood percentage

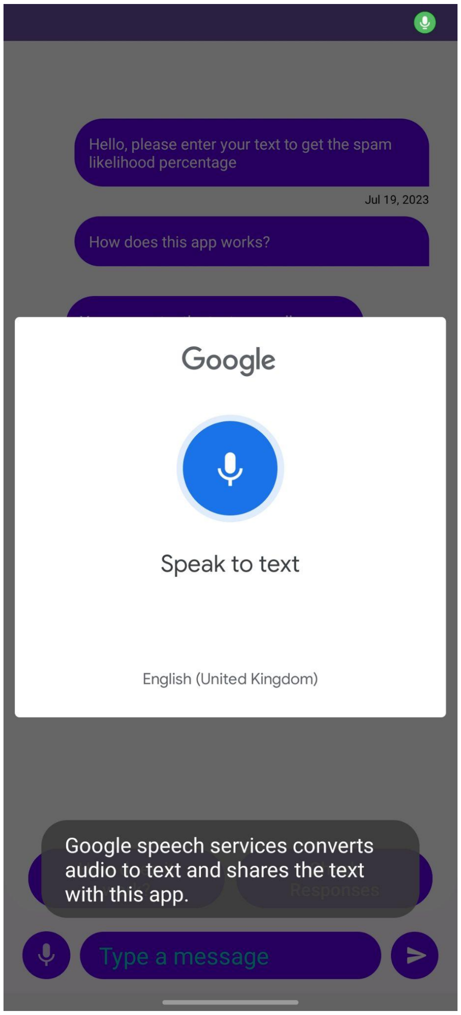
Jul 19, 2023

How does this app works?

You may enter the text manually or use the speech to text feature by clicking on the record button on the bottom left. We will then take the text entered and display the percentage of the likelihood that the text is a spam message.

3.3.2. Enter new message

Users may choose to enter text manually in the ‘Type a message’ box or click on the button at the bottom left corner with a microphone icon. Upon clicking the button, it will pop up the following which make use of google speech service to convert audio to text. Our application will then run the python program implemented into the application and display the likelihood of the text being a spam message in the form of percentage.



Hello, please enter your text to get the spam likelihood percentage

Jul 19, 2023

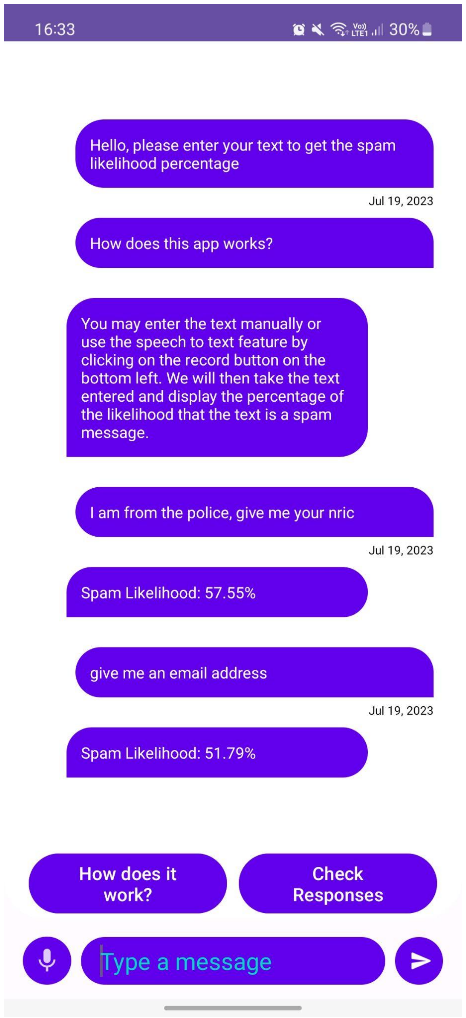
How does this app works?

Google

Speak to text

English (United Kingdom)

Google speech services converts audio to text and shares the text with this app.



16:33 30%

Hello, please enter your text to get the spam likelihood percentage

Jul 19, 2023

How does this app works?

You may enter the text manually or use the speech to text feature by clicking on the record button on the bottom left. We will then take the text entered and display the percentage of the likelihood that the text is a spam message.

I am from the police, give me your nric

Jul 19, 2023

Spam Likelihood: 57.55%

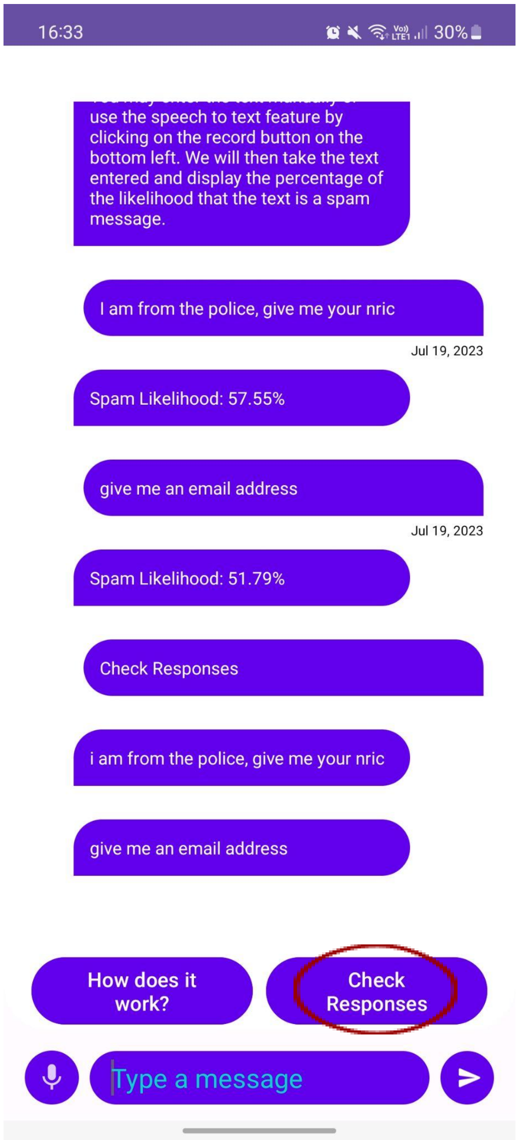
give me an email address

Jul 19, 2023

Spam Likelihood: 51.79%

3.3.3. View previous messages

After inputting multiple messages, users can click on the ‘Check responses’ button to get their previously entered messages.



use the speech to text feature by clicking on the record button on the bottom left. We will then take the text entered and display the percentage of the likelihood that the text is a spam message.

I am from the police, give me your nric

Jul 19, 2023

Spam Likelihood: 57.55%

give me an email address

Jul 19, 2023

Spam Likelihood: 51.79%

Check Responses

i am from the police, give me your nric

give me an email address