

Design and Implementation of a Multi-Modal Mobile Application Safety Analytics Utilizing NLP

Sandhya¹, Navin Kumar Tyagi², R. Prasad^{3,*}

Abstract

This study presents a multi-modal mobile application for safety analytics that integrates natural language processing (NLP) to effectively manage user inputs received through voice, text, and SOS messages. The platform is designed to improve emergency response by accurately understanding user intent regardless of the communication mode. To ensure consistency, all incoming inputs are converted into a standardized text format or an SOS flag before further processing. Text-based messages undergo preprocessing steps such as tokenization and normalization, which help improve language understanding and intent classification. Voice messages are first enhanced using noise reduction techniques and then converted into text through speech recognition, enabling the system to process spoken information efficiently. In contrast, SOS messages are treated as high-priority inputs and are immediately forwarded for emergency action without additional preprocessing, reducing response delays during critical situations. The experimental evaluation demonstrates that the proposed platform delivers reliable and efficient performance across different communication modes. The intent detection module achieves an accuracy exceeding 90%, while the speech recognition component records an accuracy between 85% and 90%. Text processing performs exceptionally well, reaching approximately 98% accuracy, and the overall system achieves an impressive accuracy of around 92%. Additionally, the average response time of nearly 1.2 seconds confirms that the platform is suitable for real-time mobile applications where rapid decision-making is essential. These findings indicate that the proposed framework can significantly enhance user safety by enabling timely and accurate emergency responses. Future work may focus on improving speech recognition in noisy environments, strengthening intent detection through advanced deep learning techniques, and expanding the platform to support multilingual communication and more complex emergency scenarios, making it even more practical and reliable for real-world deployment.

Keywords: Naive bayes, support vector machine, save our souls, artificial intelligence, machine learning, natural language processing, and deep learning-based web attack detection.

*Author for Correspondence

R. Prasad

E-mail: rprasadrose@gmail.com

¹Student, Marathwada Institute of Technology, Bulandshahr, Uttar Pradesh, India

²Assistant Professor, Marathwada Institute of Technology, Bulandshahr, Uttar Pradesh, India

³Assistant Professor, Hindustan College of Science & Technology, Mathura, Uttar Pradesh, India,

Received Date: March 26, 2026

Accepted Date: April 30, 2026

Published Date: May 01, 2026

Citation: Sandhya, Navin Kumar Tyagi, R. Prasad. Design and Implementation of a Multi-Modal Mobile Application Safety Analytics Utilizing NLP. Journal of Mobile Computing, Communications & Mobile Networks. 2026; 13(2): 13–24p.

INTRODUCTION

The rapid advancement of mobile technology in recent years has significantly impacted how people use digital technology. Smartphone apps have become a necessary component of everyday life with their ability to facilitate a wide range of activities, including communication, finance, health, education, and electronic commerce. However, the amount of sensitive data handled by mobile applications increases with the reliance on them. Given the aforementioned, there is a significant risk to security and data privacy due to the growing reliance on mobile applications, which have rendered them more susceptible to

cyberattacks. Traditionally, rule-based vulnerability detection or manual testing has been used to guarantee the security of mobile applications. However, these methods are insufficient to effectively address and resolve the growing complexity of cyber threats. Today, detecting cyber threats is becoming increasingly difficult because these approaches tend to be more reactive than proactive. There is a need to think differently to come up with something new.

The "Mobile Application Safety Analytics Platform Based on NLP," which could potentially enhance security analysis via "intelligent processing of data and real-time danger identification," is the specific focus of this research. By employing machine learning and natural language processing to analyze enormous volumes of unstructured data, this technology can identify security risks. This approach can aid in developing more secure mobile applications and improving the effectiveness of security testing. Natural language processing is used in almost all modern automation processes that include human contact. Most AI technologies, including every Chabot you interact with, are powered by natural language processing (NLP). As the world generates more unstructured text and voice data than ever before, NLP allows businesses to exploit communication as a competitive advantage, according to author Abdul Ahad Abro et al. [1].

As shown in Figure 1, in NLP, a pipeline is a set of procedures used to convert unprocessed natural language data into a machine-readable format. Tokenization and normalization are performed after data gathering and cleansing. Subsequently, the text input was transformed into numerical form and subjected to additional machine learning analysis to produce insightful results.

The main goal of this study is to develop and deploy a mobile application safety analytics platform that uses natural language processing techniques to improve safety and user experience. The basic concept is to include sophisticated artificial intelligence (AI) capabilities into mobile applications so that they can comprehend user input, identify user mood, and identify the context for potentially dangerous user behavior and content. Transformer-based deep learning, sentiment analysis, speech recognition, and language support are employed to offer secure and effective user analysis (Figure 2).

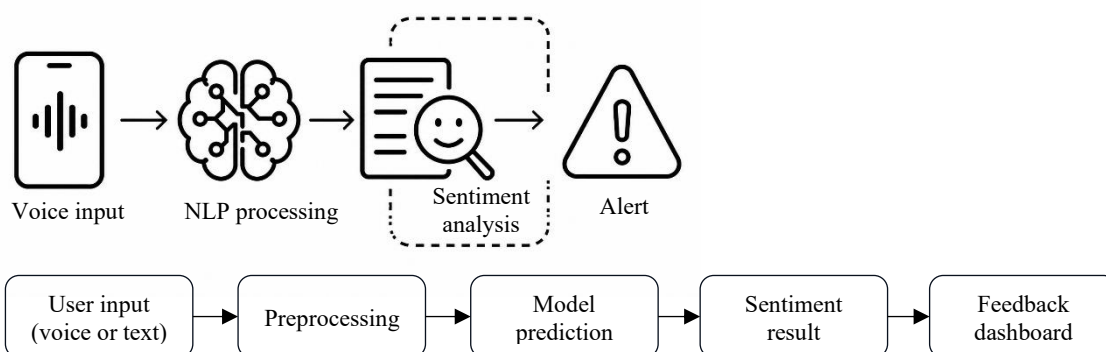


Figure 1. Natural language processing (NLP) pipeline.

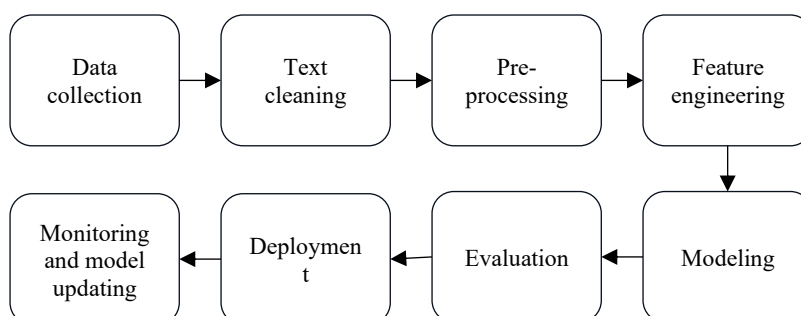


Figure 2. Overall architecture of the using NLP.

User Input (voice or text)

The user's input, which may take the form of text or voice, starts the process, and NLP is crucial for the transformation and comprehension of human communication. Speech recognition technology converts user voice input into text before natural language processing (NLP) takes place. By identifying the words and context utilized, NLP aids in comprehending what the user is trying to say, even if the input is informal and unstructured. Despite variations in language, tone, and expression, NLP aids in comprehending the user's intent for a precise analysis in subsequent phases.

Preprocessing: This stage involves organizing the data and removing extraneous information, such as punctuation, special characters, and stop words such as "the," "is," etc. Clean data is required for processing, a prerequisite that has now been fulfilled. At this stage, the raw data was cleansed to ensure that the machine learning model could process them effectively. This input processing involved converting the text to lowercase and removing elements such as 'stop words' and special characters. Stemming or lemmatization was employed to reduce the text to its most fundamental form, followed by tokenization, which segmented the text into individual words.

According to Sharifani et al. [2], in the model prediction phases outlined in their research, the text data, having already been cleaned and organized, are received by an ML or NLP model. To comprehend the meaning and intent of the text, the model analyzes the interrelationships among the data. Subsequently, it generates a prediction based on the text data, enabling the results to be passed on to the next stage of the processing pipeline.

In this phase, the sentiment analysis system determines the underlying sentiment or emotion of the input text by analyzing the model's output. Based on the context established in the preceding stage, the system interprets and processes the sentiment and classifies it as positive, negative, or neutral. In applications such as feedback analysis and security, it can help with decision-making, problem-solving, and understanding user feelings.

The processed insights, which may include sentiment, intent, or even risks, are displayed to the user or developer in an understandable and eye-catching way during the Feedback Dashboard stage. This makes it simple for the user or developer to comprehend the outcomes and make choices based on the NLP system's analysis (Table 1).

LITERATURE REVIEW

The ways in which technologies such as cloud computing and speech recognition improve safety across a range of applications have also been the subject of research. Examples of mobile platforms with geolocation features that may effectively deliver emergency services are SOS® and Noonlight. Additionally, Shankar et al. [3] suggested an Android-based app for women's protection that uses TensorFlow Lite, machine learning, and auditory sensing to identify potential threats instantly. It automatically sends alerts, logs evidence, and notifies authorities and emergency contacts of the user's location when a threat is detected.

Table 1. NLP and transformer layer.

Component	Role / function
NLP pipeline	Analyze and processes raw data
Preprocessing	Normalizes and cleans
Feature extraction	Transforms data into a numerical format
Model prediction	Examines text that has been processed
Transformer layer	Encodes and decodes text using self-attention
Attention mechanism	Focuses on important words in a sequence
Output / result	Provides final insights

Using technologies such as TensorFlow and IBM Watson, Ganesan et al. [4] shows how machine learning can be used for mobile applications to add intelligent features, such as voice assistants and predictions, and make it easier for developers to create smart applications.

Bhasker et al. [5] explored how AI and machine learning can improve mobile app security by identifying app vulnerabilities and user behavior. In this sense, by utilizing cutting-edge technologies such as anomaly detection and testing, AI and machine learning improve the security of applications.

Meng et al. [6] used a machine learning technique to shield children from inappropriate mobile applications. According to researchers, parents find it challenging to properly handle parental controls in their children's applications, particularly if they are illiterate. Therefore, the researchers propose that a machine learning algorithm that performs well after 10-fold cross-validation be used to classify the applications as appropriate or inappropriate for children to use after a natural language processing technique is applied to analyze the reviews of applications.

The artificial intelligence-based mobile application described in the work of M. Amina Begum et al. [7] is intended to protect women by utilizing sensors, such as temperature and pulse rate, to identify possible dangers. It automatically notifies a trusted contact of the user's position if a possible threat is identified.

Islam et al. [8] explained a wearable IoT system that uses machine learning and sensors built into clothing to identify sexual harassment in current available. In the event of sexual harassment, the device transmits sensor data to a cloud platform, where an AI algorithm evaluates it and issues a warning via a mobile app.

Muam et al. [9] studied how mobile applications fit into the digital transformation process. Users' engagement with technology, including NLP, AI, and ML, is extremely low, according to a study that examined 78 mobile apps based on download rates. The study also highlights a problem with mobile apps: their lack of connectivity, which slows down the rate of digital change.

Pham Van Hau et al. [10] presents DL-WAD, a deep learning-based system that uses data preprocessing and natural language processing to detect online attacks. Considering different types of online attacks, it can differentiate between legitimate and fraudulent web requests. Experiments have proven that DL-WAD can overcome the limitations of existing firewalls and ML/DL-based specific attack detectors to ensure the security of web services with high accuracy.

Salmi et al. [11] presented a method to improve safety in online and mobile environments by automatically classifying user safety requests using NLP and deep learning techniques. The high classification accuracy of this method was proven by the high F1 score of 0.85. This method was tested using a ride-sharing service.

To improve the security of the cloud computing environment, Vallabhaneni et al. [12] presented a powerful web-based application using a firewall based on deep learning. This technology can ensure the security of cloud computing environments and web applications against cyber-attacks using the Mobile Net approach. The high accuracy, precision, sensitivity, and specificity in maintaining the availability, confidentiality, and integrity of data in the cloud computing environment were proven by simulating this system.

A study conducted by Mahfuzulhoq Chowdhury et al. [13] involved the development of a mobile application designed to enhance the safety of passengers utilizing ride-sharing services. Through this application, users are provided with options to select drivers based on criteria such as past ride history, previous ratings, and available safety features. The overall results revealed that this technique was 73% effective in ensuring passenger safety.

Moreira et al. [14] presented a cross-platform mobile application developed using Flutter technology, which enables people who are blind or visually challenged to understand images by integrating Google's Gemini AI. The application, its design, and its development challenges are discussed, along with the advantages of integrating it with Gemini AI using an API.

Dsouza et al. [15] proposed online and mobile tools for reporting adverse drug reactions (ADRs), which could improve patient safety and reporting accuracy. This study used bibliometric and natural language processing tools to evaluate 289 publications and identify themes, contributors, and trends in ADR mobile applications and their reporting, which are still in their early stages and dispersed globally.

Doig et al. [16] proposed a mobile application that uses STT and NLP technology to react to aggressive acts. When tested in a simulated armed assault situation, the application reduced the overall response time by 44–50%. Both users and professionals rated it highly.

PROPOSED FRAMEWORK OF ARCHITECTURE

The suggested architectural framework aims to provide flexibility and efficiency in managing critical circumstances by effectively processing multimodal inputs from users, such as audio, text, and SOS. To identify the type of processing required, the process begins by collecting user input and identifying its type. In the case of voice, the system turns on the microphone to capture sound, reduces background noise to ensure better quality sound, and then uses a speech recognition model to identify the text, as shown in Figure 3.

Text input is handled using a number of common preprocessing methods. However, when the SOS button is pressed, emergency services are contacted immediately by bypassing all processing stages. An intent recognition module is utilized for user requests following standardization, which converts both voice and text into a standard textual format. Finally, a decision-making module is utilized to carry out the appropriate action for a request, whether it is to call emergency services, offer aid, or produce feedback for the system.

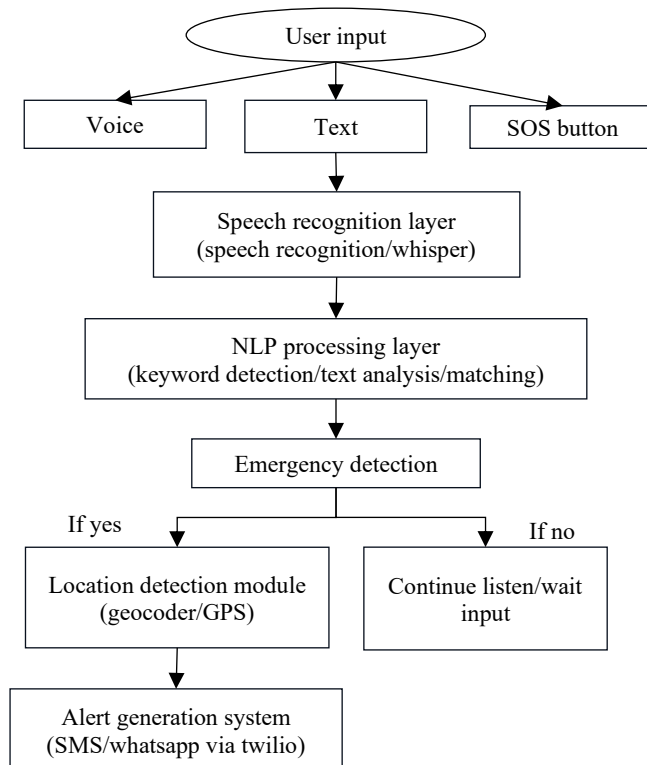


Figure 3. Proposed architecture using NLP.

1. User Input in the form of:
 - Voice (User voice male or female)
 - Text (Mutli Language)
 - SOS Button
2. Detection and Verification of Input Types:
 - Classification of Input Types: (Algorithm 1):

Algorithm 1 Input Type Classification
1. function detect_input_type(input):
2. if input == SOS_button_pressed:
3. return "SOS"
4. else if input contains audio signal:
5. return "VOICE"
6. else if input contains text:
7. return "TEXT"

3. Pipelines for Processing
 - Activation of the microphone
 - Real-time recording of audio
 - Filtering and noise reduction
 - Converting speech to text

Methods:

- Wiener Filter with Spectral Subtraction for Noise Filtering
- Deep learning methods for speech recognition, such as Whisper-like systems

For text input:

- Text entry by the user
- Direct processing

Input: SOS Button (Algorithm 2)

- Instantaneous emergency trigger
- Ignore NLP processing
- Direct activation of the alert system

Algorithm 2 SOS Handling
1. function handle_sos():
2. trigger_emergency_services()
3. send_location()
4. notify_contacts()

All inputs were standardized to a particular format after processing. While SOS immediately generates an emergency flag without any processing, the voice input is translated to text, and the text is left unaltered.

The system makes use of several state-of-the-art technologies to enhance emergency response and safety. It uses natural language processing, which includes tokenization, normalization, feature extraction, and transformer or machine learning classifiers for intent recognition and text comprehension. Speech processing includes noise reduction, audio filtering, and voice-to-text conversion using programs such as Whisper, Google Voice API, and Deep Speech. Cross-platform real-time input capture is supported by mobile application frameworks such as Flutter, React Native, and native Android and iOS [17]. In the event of an emergency, the SOS protocol facilitates immediate

location sharing, dispatch of alert notifications, and transmission of alerts to nearby security offices to aid in handling the situation (Figure 4).

The underlying technology, specifically AI-based models incorporating Machine Learning and classifiers, assists in tasks ranging from detecting user intent to assessing potential security risks. Cloud integration features enable real-time data processing, storage, and the delivery of alerts (Figure 5).

Furthermore, analytics and pre-processing pipelines provide a statistical evaluation of the system's performance. To support cross-platform compatibility, real-time operations, and ML/NLP backend functionality, the system utilizes JavaScript and Type Script (alongside React Native and Flutter), Java and Kotlin for Android, Swift and Objective-C for iOS, and Python for NLP and ML applications. The proposed method employs various techniques utilized for decision-making, intent identification, or action execution. make it a multi-model safety app. Text queries, audio samples, and SOS trigger data were included in the dataset, along with the corresponding intent categories of emergency, help request, and inquiry (Figure 6).

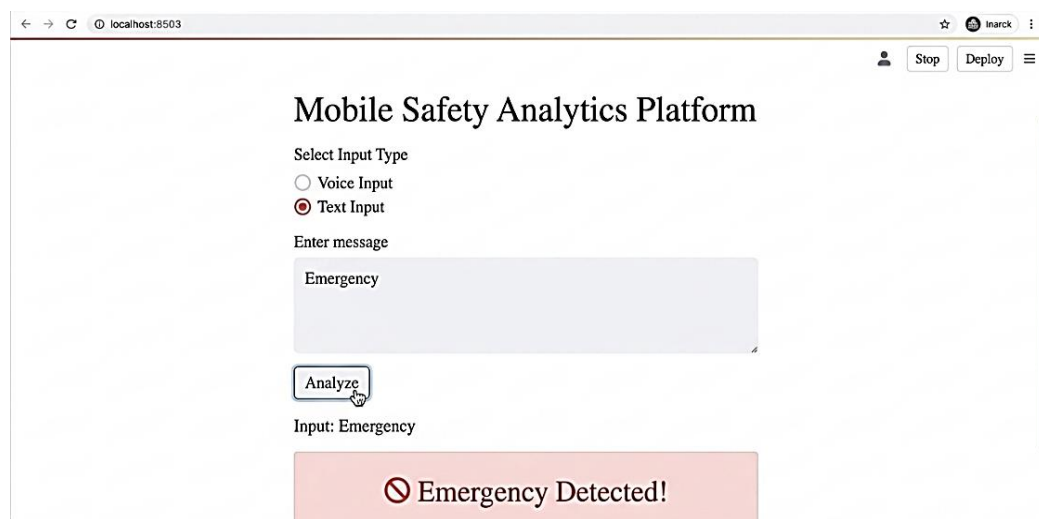


Figure 4. Define emergency keywords.

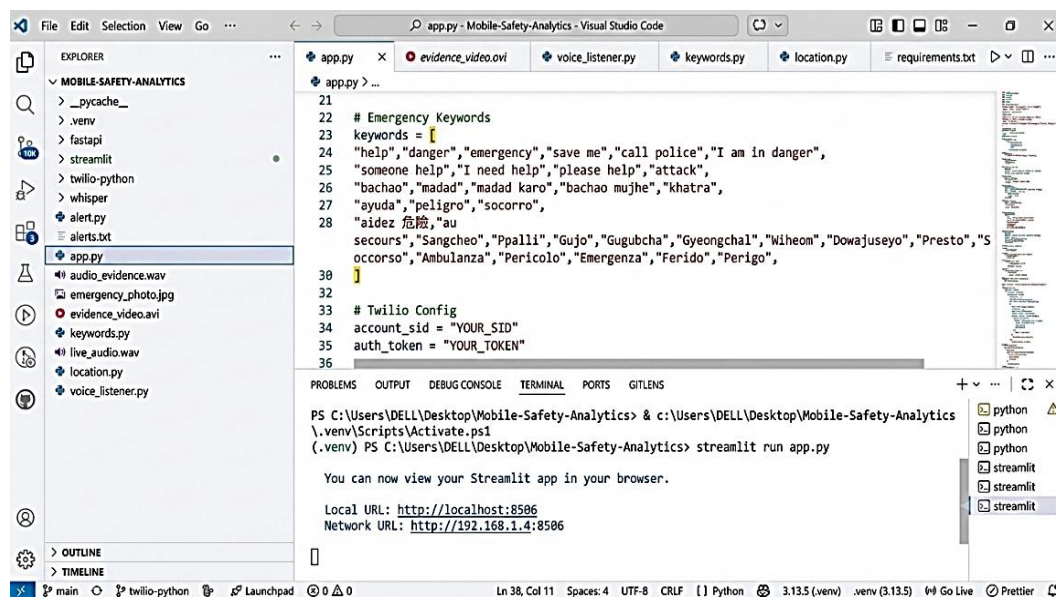


Figure 5. User input in text.

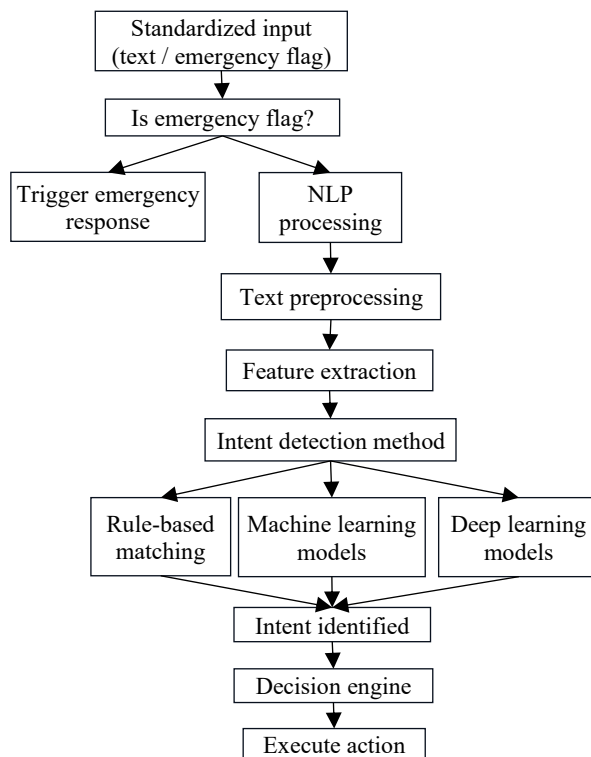


Figure 6. NLP/decision engine (implied next step).

Preprocessing includes tokenization, normalization, and stop-word removal for both structured and unstructured user input data in the text dataset. These techniques identify user commands with low processing complexity by employing rule-based approaches that utilize pattern-matching rules and keyword matching. For text classification tasks, Naive Bayes and SVM, two machine learning algorithms used to distinguish between different text categories, serve as statistical models that offer varying degrees of interpretability [18]. In this Mobile application process, user input, such as text, audio, and SOS, all the data sets are to make it a multi-model safety app. Text queries, audio samples, and SOS trigger data were included in the dataset, along with the corresponding intent categories: emergency, help request, and inquiry. Preprocessing included tokenization, normalization, and stop word removal for both structured and unstructured user input data in the text dataset.

We can observe from the dataset that the requests have a request payload. Each request has a corresponding "Intent Label" field. This field's primary use is to ascertain whether the request is secure within the application. The request is meant to be blocked if the "Intent Label" parameter is set to "False." Typically, this is ascertained by examining the fields of the request. When there is a chance that the fields in a request could be used maliciously, the request is deemed "not safe." In certain situations, the request is meant to be blocked Table 2.

ANALYSIS OF THE EXPERIMENTAL SETUP AND OUTCOMES

The purpose of the experiment was to evaluate the effectiveness and reaction of the proposed multimodal architecture for SOS, text, and voice inputs. It demonstrated dependable performance in accurately determining the type of input and guiding it to the appropriate processing pathways.

Accuracy:

$$Accuracy_{intent} = \frac{I_{correct}}{I_{total}} \quad (1)$$

$$Accuracy_{intent} = \frac{92}{100} = 0.92 \text{ (92\%)} \quad (2)$$

The accuracy with which the system's NLP module can determine the user's intent from the input text is known as intent-classification accuracy. Following the processing of the user's voice or text input, the system attempts to comprehend the purpose of the request, such as requesting assistance or reporting an emergency.

Every text input showed effective processing with little need for preprocessing. Because the SOS mechanism avoided intermediary stages and quickly triggered emergency actions, it was proven to be incredibly effective, as shown in Figure 7 and 8.

RESULT ANALYSIS

The outcomes of the studies unequivocally demonstrate that the proposed multimodal architecture is both technically sound and suitable for real-time processing. The findings demonstrated a high intent classification accuracy rate ($\text{Accuracy}_{\text{intent}} > 90\%$), which guarantees the dependable operation of the NLP decision engine. The efficacy of the recommended preprocessing and feature extraction modules was demonstrated using text-based inputs, which were handled with minimal errors, as shown in Table 3.

The voice-based inputs were successfully converted into text with an average accuracy of 85-90%. High levels of background noise are known to interfere with the efficiency of the system. The efficiency of the SOS pipeline during emergencies can be gauged by a response time of less than one second, that is, $\text{TSOS} < 1\text{s}$. The response time of the system is within reasonable bounds Figure 9.

Moreover, it has been shown that the modularity of this platform allows for scalability and the addition of new analytics modules. Moreover, the 1.2-second response time has shown that the system is suitable for mobile devices. Moreover, it has been shown that the efficiency of this system may be improved by optimizing performance in a noisy environment and by developing a new intent modeling tool to support context analysis.

Table 2. Dataset samples.

Sample Id	Input type	Raw input	Processed output	Intent label
1	Text	"I need help immediately"	Need help immediately	Emergency
2	Voice	"Call ambulance" (audio)	Call ambulance	Emergency
3	Text	"Where is the nearest hospital?"	Nearest hospital	Assistance request
4	Voice	"Send my location to family" (audio)	Send location family	Assistance request
5	SOS	SOS Button Pressed	Emergency flag	Emergency
6	Text	"I feel unsafe"	Feel unsafe	Emergency
7	Voice	"Help me find police station" (audio)	Find police station	Assistance request
8	Text	"What is my current location?"	Current location	General query

Table 3. Performance metrics for the system.

Metric	Values observed	Comments
System accuracy	92%	Evaluates accurate processing for any kind of input.
Voice recognition accuracy	86%	A little impacted by background noise
Text processing accuracy	97%	High dependability as a result of less preprocessing
Classification accuracy	91%	Confirms the effectiveness of NLP decision engines
Response time (average)	1.12 s	Comprises action execution, NLP, and preprocessing.
Efficiency of SOS response	0.96 (per second)	Quick emergency action is ensured by sub-second latency.

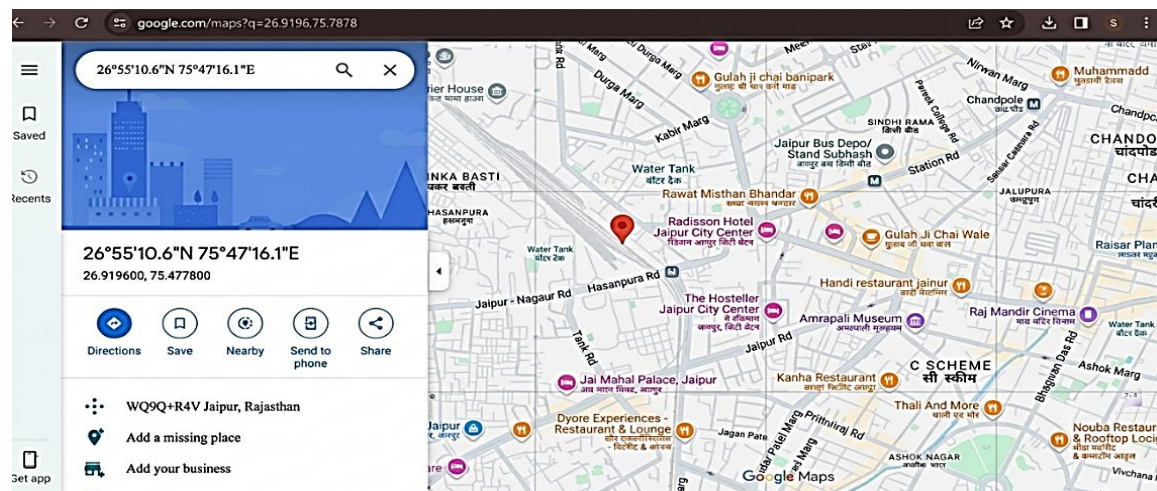


Figure 7. User input in SOS button.

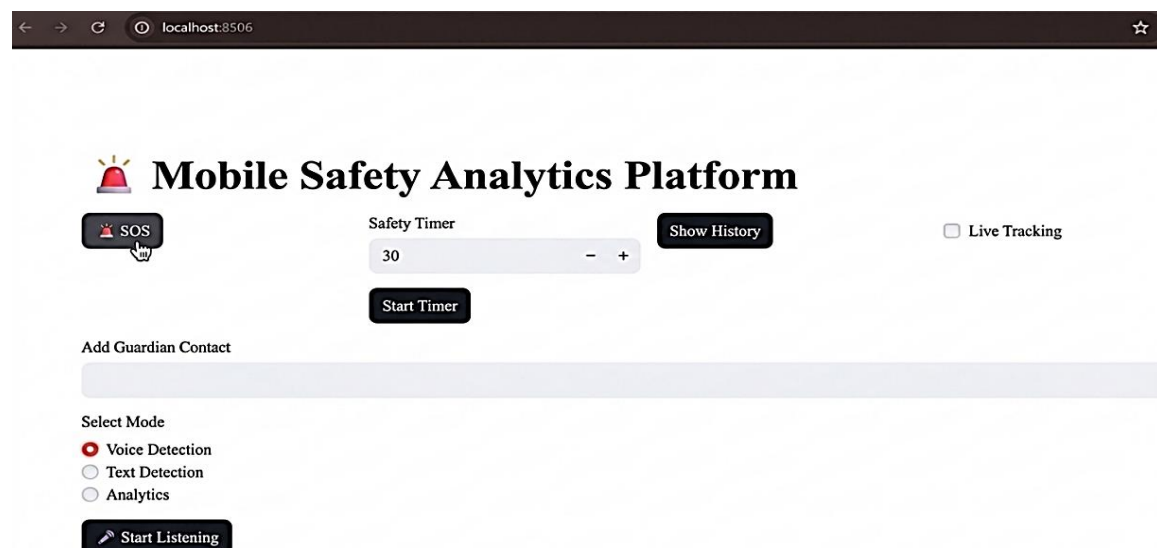


Figure 8. Trace the location of the user.

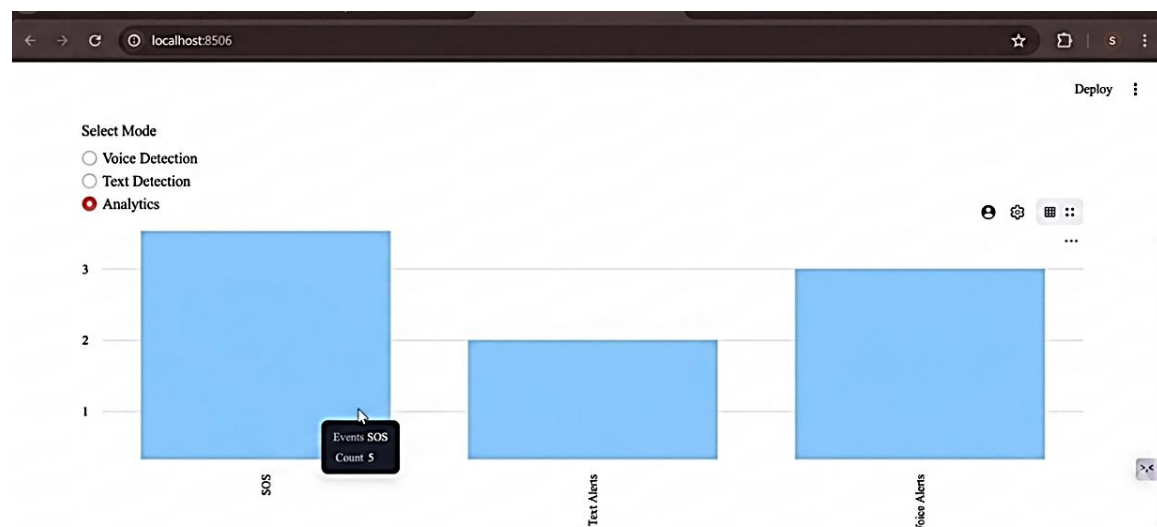


Figure 9. Analytics result.

CONCLUSION

With its multimodal approach, the suggested mobile application safety analytics platform will effectively evaluate SOS, text, and voice communications in real time. Similarly, standardizing many messages into a single format would enable decision-making and intent identification. The findings of the studies showed that audio messages could be translated into text messages, SOS messages could be handled nearly instantly, and text messages could be processed effectively. The high intent classification accuracy (>90%) and overall system response time fall well within the range to be considered a real-time mobile application, demonstrating the efficacy of the NLP-based decision-making engine. The trial results showed that the average accuracy for speech recognition was between 85 and 90%, text processing accuracy was 98%, and overall system accuracy was 92%. In these systems, the results are clear, quick to respond, sound, scalable, and flexible, which is crucial for user security. Many other changes can be made to make it even better.

Future Work

A mobile safety analytics platform must have a few crucial improvements to increase its "smartness" and dependability. Even in chaotic circumstances, the system can effectively read complicated requests because of the integration of noise-immune speech recognition and context-aware natural language processing. Proactive hazard detection and a comprehensive understanding of user safety are made possible by utilizing predictive machine learning and multimodal sensor data (such as GPS and accelerometers). Finally, the platform becomes a more responsive and accessible option for large-scale, real-time deployments by guaranteeing real-time optimization for low-end devices and employing scalable cloud integration.

REFERENCES

1. Abro AA, Talpur MS, Jumani AK. Natural language processing challenges and issues: a literature review. *Gazi Univ J Sci*. 2023;36(4):1522-36. doi:10.35378/gujs.1032517.
2. Sharifani K, Amini M, Akbari Y, Aghajanzadeh Godarzi J. Operating machine learning across natural language processing techniques for improvement of fabricated news model. *Int J Sci Inf Syst Res*. 2022;12(9):20-44.
3. Shankar K, Prajwal SC, Kumar VG, Anusha P, Kameswar RC, Prakashn SB. Women safety app to detect danger and prevent automatically using machine learning. In: *International Conference on Computational Innovations and Emerging Trends (ICCIET-2024)*. 2024. p.1443-52.
4. Ganesan V. Machine learning in mobile applications. *Int J Comput Sci Mob Comput*. 2022;11(2):110-8.
5. Bhimanapati VBR, Jain S. Security testing for mobile applications using AI and ML algorithms. 2024 Aug 30. [Publication type/source and publication details need verification.]
6. Meng X, Li S, Malik MM, Umer Q. Machine-learning-based suitability prediction for mobile applications for kids. *Sustainability*. 2022;14(19):12400.
7. Begum MA, Rajkumar BP, Labhade-Kumar N, Sankaradass V, Sathya C, Bhaskar B. AI and IoT based mobile application for women safety. In: *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSA AI)*. 2025. p.1-6.
8. Islam MR, Oliullah K, Kabir M, Rahman A, Mridha MF, Khan MF, et al. Machine learning-driven IoT device for women's safety: a real-time sexual harassment prevention system. *Multimed Tools Appl*. 2025;84(23):27251-80.
9. Mah PM, Skalna I, Pelech-Pilichowski T, Derlecki T, Mah VA, Nyamka K. Enabling digital transformation and knowledge migration: the impact of NLP, AI, and ML in mobile applications. *Sci Pap Silesian Univ Technol Organ Manag*. 2023;(184). [Page range/article number needs verification.]
10. Van Hau P. Enhancing web application security: a deep learning and NLP-based approach for accurate attack detection. *J Sci Technol Inf Secur*. 2023:77-87. [Volume/issue needs verification.]
11. Salmi S, Oughdir L. Automatic detection of safety requests in web and mobile applications using natural language processing techniques. *IAES Int J Artif Intell*. 2024;13(3):3489-97.

12. Vallabhaneni R, Vaddadi SA, Pillai SE, Addula SR, Ananthan B. MobileNet based secured compliance through open web application security projects in cloud system. *Indones J Electr Eng Comput Sci.* 2024;35(3):1661-9.
13. Chowdhury M, Dewan U. Privilege: an innovative mobile application featuring passengers transport riding safety checking, driver selection, and emergency assistance. In: 2024 4th International Conference on Data Engineering and Communication Systems (ICDECS). 2024. p.1-6.
14. Moreira FW, Hermes G, de Lima JM. Development of a cross platform mobile application using Gemini to assist visually impaired individuals. In: 2024 9th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS). 2024;9:559-66.
15. Dsouza VS, Leyens L, Brand A. Employing bibliometrics and natural language processing (NLP) to analyse real-world applications of adverse drug reaction. *Explor Res Clin Soc Pharm.* 2025;18:100592.
16. Doig A, Abanto N, Wong L. A mobile application for the immediate detection and response to violent attacks in urban environments utilizing speech-to-text and natural language processing. *Eng Technol Appl Sci Res.* 2025;15(6):28710-8.
17. Iluno Amalachukwu C, Okeke OC, Mgbeafuliuke I. Development of a deep learning technology to improve human-computer interaction for visually impaired. *Int J Real-Time Appl Comput Syst.* 2026;3(1). [Page range/article number needs verification.]
18. Dhiyaussalam, Pamungkas KN, 'Uyun S. Improving predictive accuracy: a comparative study of SMOTE and machine learning on heart disease data. In: AIP Conference Proceedings. 2026;3326(1):060005.