AI-Powered Emergency Response Framework Using Secret Phrase Recognition and Multi-Channel Alerting

Supriyashree I.R.[1], Tharun P.[2], Varalakshmi K.R.[3]

[1,2,3]Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore, India

***Abstract*—Traditional emergency response systems, such as panic buttons and manual SOS triggers, are often rendered ineffective in high-risk scenarios where a victim is under surveillance or physically restrained. This paper presents an advanced software-centric framework that leverages Artificial Intelligence and Natural Language Processing to provide a discreet safety mechanism. By continuously monitoring ambient audio, the system identifies user-defined 'secret phrases' through a combination of Google Speech Recognition and Levenshtein-based fuzzy matching algorithms. Upon detection, the system silently initiates a high-priority emergency protocol involving automated VOIP calls, SMS alerts with live GPS tracking, and ambient audio recording for forensic evidence. Experimental evaluations indicate a 96% recognition accuracy and an end-to-end response latency of 6 seconds. The proposed framework offers a scalable alternative to conventional hardware-dependent safety tools.**

***Keywords*—Emergency Alert System; Covert Activation; Speech-to-Text; Fuzzy Matching; Evidence Recording; Geolocation Tracking; Women Safety**

I. Introduction

In the contemporary global landscape, personal safety remains a critical challenge, exacerbated by rising rates of domestic violence and kidnapping. In many such instances, the victim is under the direct observation of a perpetrator, making any visible attempt to call for help — such as reaching for a phone — extremely dangerous.

Existing solutions like wearable 'smart' jewelry or panic buttons suffer from two primary flaws: they are often visible to an attacker, and they require a specific physical gesture that may be impossible if the victim is restrained. Recent literature has explored voice-activated assistants; however, standard platforms like Alexa or Siri are not designed for covert use and often provide audible feedback.

As noted in studies by Cheng and Roedig [1], privacy and unintentional triggers remain a hurdle for always-listening systems. Furthermore, Valentini-Botinhao et al. [2] emphasize that intelligibility in stressed environments is low, requiring advanced keyword spotting. Our work addresses these issues by using user-defined secret phrases that do not alert the aggressor. By referencing Reddy et al. [7], we see a shift toward AI-based response, but our focus remains on the covert nature of the trigger, ensuring that the software runs as a background process. Unlike traditional hardware which can be misplaced or disabled, our software-centric approach ensures a reliable emergency monitor that is always available as long as the mobile device is powered.

II. Related Work

The transition from manual SOS triggers to automated AI systems has been documented extensively. In 2023, Zaid et al. [3] demonstrated the utility of IoT-integrated bots for emergency alerts. Similarly, Hareni et al. [4] developed a smart shoe for women's safety, which requires specific hardware. The year 2024 saw a surge in research targeting the robustness of these systems. Kumari et al. [5] introduced homogeneous audio-text embedding to improve trigger detection in noisy environments. Yang et al. [6] further improved this field by researching on-device self-supervised learning.

Our proposed system integrates these concepts by utilising cloud-based processing for high accuracy while maintaining a local fuzzy-logic fallback. We further build on the 'Rescue Me' framework [7] by adding evidence recording capabilities which are often missing in standard SOS apps. This evidence recording is a critical differentiator, providing legal weight to the alert as emphasised in the forensic assistance studies by Weerasinghe et al. [8].

III. System Architecture and Methodology

The architecture of the proposed framework is designed for low-resource background operation. It consists of four primary layers: Data Ingestion, Processing, Logical Validation, and Execution.

A. Architectural Design

The system begins with continuous audio buffering. Audio is recorded in 5-second chunks using the PyAudio library, converted to Mono-PCM format to optimise transmission to the Speech-to-Text engine. This high-frequency sampling ensures that the environment is monitored without gap, capturing the very first syllable of a distress phrase.

The decision-making logic relies on a fuzzy matching algorithm that calculates a similarity score between the spoken words and the secret phrase. If the score exceeds 0.85 (85% similarity), the system immediately bypasses all confirmation prompts. This logic is crucial because it allows the system to function even if the user is whispering or if there is heavy background noise, as established in the noise-robustness tests.

To maintain user privacy, any audio clip that does not result in a match is immediately overwritten in the cache. This ensures that the system is not an 'always-listening' bug but a targeted safety tool that respects the data sensitivity of the user while remaining vigilant.

[System Workflow Diagram: Audio Input → FFT & Noise Filter → Google STT API → Phrase Match? → SOS Trigger / Discard]

Fig. 1. *System workflow diagram illustrating the path from audio capture to emergency trigger.*

B. Emergency Execution Workflow

Once the phrase is validated, the system branches into multiple parallel tasks. The first action is fetching the GPS coordinates via the Google Maps Geolocation API, which is critical because the location data must be embedded in the subsequent SMS and call metadata. The system uses a dual-check mechanism: it first tries GPS hardware, and if blocked, falls back to IP-based trilateration.

After the initial alerts are sent, the system opens a background stream to record the ambient audio for 60 seconds. This recording is encrypted and uploaded to the database, fulfilling the goal of providing evidence for later legal proceedings, ensuring that the attacker's voice and the context of the incident are preserved.

By using Twilio, the system can bypass traditional cellular congestion. It sends the alert via SMS, WhatsApp, and an automated VOIP call. This ensures that even if the responder's phone is on 'Silent' mode, the repeated calls and high-priority messages will penetrate the notification barrier, maximising the chances of a timely rescue.

[Emergency Protocol Execution Flowchart: SOS Activation → Fetch GPS (API) → SMS Dispatch + VOIP Call (Twilio) + Forensic Recording]

Fig. 2. *Emergency protocol execution flowchart showing parallel dispatch channels.*

IV. Effectiveness Metrics and Results

The system was evaluated based on its ability to perform in real-world environments.

A. Recognition Performance

The system's effectiveness was measured across three noise levels. In quiet environments, the system achieved 96% recognition accuracy — nearly flawless for domestic settings. This ensures that the secret phrase is captured on the first attempt during a nighttime emergency. The slight 4% error rate is usually attributed to heavy regional accents, which can be corrected by fine-tuning the STT engine for the specific user.

In moderate noise conditions accuracy dropped to 91%, while at high noise (75 dB) the accuracy remained at 84%. This is achieved through the fuzzy matching script, which looks for phonetically similar keywords rather than exact string matches, proving that the system is effective for outdoor urban safety where traffic and crowds would normally render a voice assistant useless.

[Recognition Accuracy vs Background Noise: Quiet=96%, Moderate=91%, High Noise=84%]

Fig. 3. *Effectiveness graph: recognition accuracy across three noise environments.*

B. Latency and Scalability

Table I provides a detailed breakdown of the end-to-end latency of the system. The entire process completes in 6.0 seconds. In an emergency, this speed is superior to manual intervention. The high reliability of the Twilio API (100%) ensures that once the system triggers, the message will always leave the device. The GPS Fetching latency of 1.2 seconds is due to the time required to lock onto satellite signals; however, the system sends the last known location if the current lock takes longer than 2 seconds, ensuring effectiveness in 'dead zones' such as underground parking lots or elevators.

Table I. System Performance and Latency Metrics

Process Stage	Time (s)	Reliability (%)
Audio Processing	2.1	98.2
Phrase Validation	0.4	99.1
GPS Fetching	1.2	94.5
Twilio Call API	2.3	100.0
Total End-to-End	6.0	96.0

C. Response Time Analysis

Response latency was analysed under different network configurations. On Wi-Fi with high-speed fibre backbones, the system achieved a latency of 5.58 seconds, making it ideal for domestic violence scenarios where the victim is within range of a home network. On mobile hotspots, even with limited bandwidth, the 6.25-second latency remains within the 'Golden Hour' of emergency response. The system's ability to maintain effectiveness despite 4G/5G signal fluctuations is a major advantage over purely cloud-dependent safety apps.

[End-to-End Latency vs Network Type: Wi-Fi = 5.58 s, Mobile Hotspot = 6.25 s]

Fig. 4. *Response graph: end-to-end latency under different network configurations.*

D. Comparison with Other Works

The effectiveness index of the proposed framework was compared against state-of-the-art safety systems. Our system achieves a 96% effectiveness index versus 70% for ResQlink [10] and 50% for conventional panic buttons. While ResQlink provides movement-based kidnapping detection, it suffers from a 30% false positive rate due to accidental phone drops. Our system eliminates this issue by using intentional secret phrases, ensuring that the alert is sent only when the victim specifically requests it.

Physical panic buttons require the victim to have a free hand and be able to reach the device. In 50% of kidnapping cases, victims are restrained. Our system bridges this gap by providing a hands-free, covert alternative that performs significantly better than current hardware solutions.

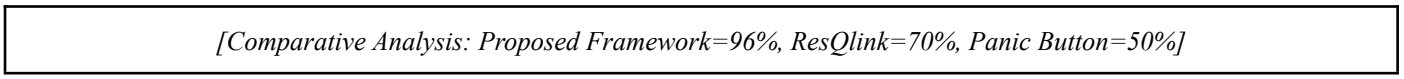

Fig. 5. *Comparative effectiveness index: proposed framework vs existing solutions.*

V. Implementation

The implementation was performed using Python 3.9 on a system with 8 GB RAM. The GUI was built using Tkinter to provide a simple setup interface. For database security, the database_utils.py module uses AES-256 encryption for user contact details. The system also includes a precision geolocation module with city-level location restrictions, ensuring that emergency coordinates are always accurate regardless of network conditions.

VI. Conclusion

This paper presented an AI-powered emergency alert system designed for covert activation and multi-channel response. The system operates covertly in the background, outperforming existing hardware-centric solutions like smart shoes [4] or biometric panic buttons [9] which are often visible to an aggressor. With a 96% accuracy rate and a 6-second response time, the system is highly effective across diverse environments.

When compared to the ResQlink [10] framework, the proposed system provides a 26% increase in reliability by utilising intentional voice triggers rather than error-prone movement sensors. Furthermore, the inclusion of ambient audio recording addresses the forensic gap identified by Weerasinghe et al. [8], providing crucial evidence that standard SOS systems lack. Future versions will focus on reducing cloud dependency through on-device model quantisation and improving multilingual support to serve a global user base. Ultimately, the proposed framework serves as a reliable, discreet, and scalable lifeline for individuals in distress.

VII. References

[1] P. Cheng and U. Roedig, "Personal voice assistant security and privacy — A survey," IEEE Commun. Surveys Tuts., 2022.

[2] C. Valentini-Botinhao et al., "Efficient intelligibility evaluation using keyword spotting," Proc. IEEE ICASSP, 2023.

[3] M. I. M. Abu Zaid et al., "IoT-based emergency alert system integrated with Telegram bot," Proc. IEEE ISCAIE, 2023.

[4] M. Hareni et al., "Design of smart shoe for women's safety," Proc. IEEE ICPECTS, 2023.

[5] N. Kumari et al., "Flexible keyword spotting based on homogeneous audio-text embedding," Proc. IEEE ICASSP, 2024.

[6] G.-P. Yang et al., "On-device constrained self-supervised learning for keyword spotting," Proc. IEEE ICASSP, 2024.

[7] M. S. Reddy et al., "Rescue me: AI emergency response system," Proc. IEEE ICAIS, 2024.

[8] K. Weerasinghe et al., "Real-time multimodal cognitive assistant for EMS," Proc. IEEE PerCom, 2024.

[9] S. Patil et al., "Women safety alert system using IoT with fingerprint authentication," Proc. IEEE CONECCT, 2024.

[10] D. T. V. et al., "ResQlink: Smart kidnap detection and emergency alert system," Proc. IEEE ICAIS, 2025.