JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

The 'Friend in Threat' Voice Operated AI-System

Vikash Kumar[1], Tanvi Rustagi[2]

1 Student, CSE, World College of Technology Management, Gurugram, Haryana
2 Assistant Professor, CSE, World College of Technology Management, Gurugram, Haryana

Abstract

The global challenge of women's safety continues to grow because harassment incidents and urgent emergency situations both increase. Users who want safety today must activate emergency protocols by pressing buttons or by using GPS. Some real-world emergency situations do not respond well to emergency response procedures that are already in use. Users can activate emergency functions automatically via AI Voice recognition and fuzzy keyword search features in the 'Friend in Threat' application. The research develops a smart safety application by uniting NLP and ML capabilities with Android Framework to activate distress alerts which deliver pre-selected users location and identity details. This paper examines present safety applications before showing how AI voice-based interventions quicken response times during emergencies with a clear explanation of these enhancements. This paper explores detailed technical information about algorithms along with a presentation of current obstacles and future developments which include offline NLP processing and deep learning models for voice pattern recognition systems.

1. Introduction

The basic right of women to feel safe remains threatened when millions of women undergo physical dangers from domestic assault as well as sexual abuse at work and through online attacks. The fight against violence towards women has shown limited progress because 35% of women face physical or sexual violence, 7% experience sexual assault from non-partners together with nearly 200 million women who have survived female genital mutilation. India has experienced increasing violence against its women population based on statistics which show women suffer rape every 16 minutes and one-third experience domestic violence alongside ongoing dowry-related death cases. Examples of the Nirbhaya case from 2012 and the Hathras incident from 2020 and the 'MeToo' movement show how serious the problem is at revealing systemic failures in law and social acceptance of women. Security violence emerges from diverse factors which include traditional male dominance coupled with gender inequality as well as monetary dependence and poor law enforcement capabilities. The problems facing justice-seeking survivors of violence endure because the Criminal Law (Amendment) Act of 2013 and the Domestic Violence Act of 2005 coexist with ongoing obstacles such as victims not filing reports and courts taking too long and continuing to blame victims. A necessary solution requires legal improvements together with educational initiatives and technological innovations along with active community involvement to solve this national issue. The justice system benefits from specialized expedited courts along with related police and law enforcement programs which meet gender-specific needs and women require access to financial resources for escaping violence and education to prevent abuse. Modern technology now plays an essential role in protecting the safety of women because AI-powered tools such as voice-activated emergency systems and facial recognition devices combined with GPS trackers allow the prevention of crimes and aid victims. AI-based predictive analytics helps the law enforcement identify dangerous zones which enables preventive action from police departments.

The development of Android applications during these recent decades established vital safety solutions accessible to women through effective services. Mobile apps emerged as essential emergency tools during this recent decade through their ability to provide SOS buttons together with live location services and automatic alert systems to both contacts and police authorities. Real-time safety capabilities of "bSafe," "Raksha," and "Safetipin" apps result from their implementation of AI and GPS technology. Through voice

command integration these programs enable females to request assistance even without accessing their phones in threatening circumstances thus making these tools more effective for risk scenarios. Predictive analysis power of machine learning algorithms helps to detect hazardous areas and propose routes of better safety. Android applications continue to expand throughout the smartphone market to become fundamental safety tools for contemporary women because they instantly provide help while connecting victims to police agencies.

Many current safety applications have notable limitations, including slow response times, reliance on stable internet connections, and a lack of integration with advanced AI security features. To overcome these issues, we have developed an innovative women's safety app called Friend in Threat, which offers quicker, more dependable, and AI-enhanced security solutions. Unlike other options, Friend in Threat utilizes real-time voice recognition, enabling users to activate emergency alerts with specific voice commands without needing to interact with their phones manually. This feature ensures that help can be called swiftly, even if the user is unable to reach their device. Furthermore, the app includes a sophisticated location-tracking system that continuously updates the user's whereabouts and shares this information with emergency contacts and local authorities. In contrast to many apps that depend entirely on internet access, Friend in Threat also has an offline SMS alert system, guaranteeing that distress messages can be sent even in areas with weak network coverage.

Ensuring women's safety goes beyond just legal frameworks or technological solutions; it is fundamentally a societal challenge that requires a united effort from governments, communities, and individuals alike. With the ongoing advancements in technology, AI-driven safety applications such as Friend in Threat are set to play a crucial role in closing security gaps and enhancing emergency response times. The integration of voice-activated alerts, AI-based risk assessments, real-time tracking, and offline emergency capabilities positions Friend in Threat as a more effective option compared to traditional safety apps. Fostering a culture of respect, equality, and a strict stance against violence is essential for creating a world where women can live without fear.

2. Background

Women are faced with security challenges in different public and private spaces that need simple and fast emergency communication systems. Traditional security applications such as 'bSafe', 'My SafetiPin', and 'Life360' depend on user input, a situation that is not preferred in situations where the user is physically disabled or cannot use their cell phone.

3. Literature Review

The Safetipin[1] is a crowd-sourced women's security app that offers real-time safety ratings that consider lighting, crowd, and accessibility (Manazir et al. 2019). Users utilize the app through the "Track Me" feature to share their live location while also using it to submit safety audits to the platform. This application works with government entities and NGOs to strengthen urban security systems (WHO, 2021). Data from users forms the main component of this app but the system becomes less effective when user participation remains low in certain regions.

GoSafe[4] is a computer vision-based safety app that facilitates real-time crime reporting and emergency notification through direct linkage of users with law enforcement agencies (Patil et al., 2021). The incorporation of one-click SOS trigger and GPS positioning ensures quick transmission of distress signals, eliminating manual intervention-induced delays. It further incorporates audio and video evidence capture to improve report credibility while supporting the police in identifying high-risk zones. Nonetheless, its GIS-based tracking may lead to location errors, and it does not support AI-enabled voice recognition and emotion sensing for distress detection by default. Furthermore, its Aadhaar-based verification system excludes those who are without proper documentation.

The bSafe[8] app that offers real-time SOS alerts, live location sharing, and automatic video recording pertaining to the security of the user. Users can trigger emergency alerts either through voice activation or manually, which will notify authorities and their trusted contacts with live GPS coordinates instantly. Another highlight for this application is its fake call feature whereby the users can also engage in a feigned call to escape rarely pleasant situations. Moreover, automatic video and audio recording would commence in the event that an SOS signal is triggered, which would work as solid evidence during emergencies. Unlikely, it is reliant on internet connectivity so will be ineffective in places with very poor network coverage. Because the app does not feature any AI-based distress detection, users do not get alerts even when they are in distress.

FEMME [11] is a safety system based on AI that includes a wearable device and a mobile app to safeguard women using real-time tracking, emergency messages, and hidden camera detection. In an emergency situation, it assists a user in sending distress messages with GPS location, audio recording as evidence, and tracking lost devices. A radio frequency-based hidden camera detection feature is one of the distinguishing factors, which ensures a user's privacy. But it is dependent on GSM/GPS networks, keeping the system

inoperative in low-connectivity zones. Manual activation could introduce delays in responding to emergencies. Future research will involve installing AI-based voice recognition and IoT-enabled wearable integration for automatic distress sensing and proactive threat prevention.

Westmarland et al.'s (2013) critical study investigates the positioning of mobile apps as resources for safety for women in the context of domestic and sexual violence. They find that just under half of the reviewed apps serve a panic alert purpose only, and so require women to undertake a deliberate safety-seeking action, demonstrating how services can commodify women's safety. While some apps provide evidence-gathering material or informational material, practitioners raised concerns over false assurances about safety, particularly for women who are living with an abuser. Many of the apps thrived on victim-blaming frameworks, and ignored features of perpetrator's monitoring device use. The report notes that any app development should transcend a panic alert model, and include evaluations for devices, ethical design, and future commitment from app developers to work with well-meaning violence prevention experts, in order to ensure users are obtaining security, and a genuine product (Westmarland et al., 2013).

The idea of a system created by Juhitha, Pavithra, and Archana (2020) was to create a real-time mobile app directed toward enhancing women's safety and support using GPS, motion sensing with accelerometers, and emergency alert features. Users can trigger alerts with a simple shake of the phone or by pressing the volume button, sending an SMS that contains the victim's name and GPS location to registered alerts within the app. The app will also automatically call the master contact allowing for a greater likelihood of an individual being alerted quickly. The app also provides for dynamic GPS tracking, to allow for tracking while the victim is moving. The unique essence of this app is that it will send the last known location in a low battery situation, all of which can occur without an actual internet connection. While these functions are useful, a considerable absence within the system is AI-based identification of distress or voice recognition and only offers manual activation on behalf of the victim which may cause issues in circumstances where access to a phone may not be possible. An idea for the future could be enhanced functionality and the use of offline AI models and automatic voice-activated alerts (Juhitha et al., 2020).

Kataria et al. (2022) [17] describe an Android women's safety application that incorporates continuous tracking of location, live alert messaging and audio streaming for immediate support in emergencies. The application enables users of the service to record contacts for emergencies, dispatch distress messages automatically at defined time intervals by clicking a HELP button, to which messages will include either coordinates or linked maps to assist responders pinpoint the user. It allows the ability to record live audio of the user in their surrounding - and such audio could act as evidence. The various aspects are able to be triggered either manually or via voice command, but the offered service of real-time does rely on maleable connectivity context, where low connection can lead to lower reliability. The application has an offline alert setting as well. In offline alert mode, users are able to send messaging without needed the GPS, but could still warn contacts of the notion of a potential threat in low connectivity locations (Kataria et al., 2022).

Hossain et al. (2019) proposed a mobile application to address alarming numbers of gender-based violence in Bangladesh. The application aimed to protect women by providing them with features such as location tracking, communication lines to emergency contacts, emergency distress signals. The application provides an automated location service, and allows women to send alerts with one touch. The authors also stated that user experience was an important design consideration, and the full functioning of the application in many languages would lend to more access for women in Bangladesh. Although this system is useful for rapid response, the authors explored some limitations of the application. Some limitations include scaling challenges, dependent upon hardware, and lack of artificial intelligence to forecast or predict the threats to women. Future versions of the applications may include predictive analytics and increased connectivity in the cloud to provide real-time coordination with law enforcement.

The application advances women's personal security but, nonetheless, critical downsides exist, including network dependency, no AI automation, manual activation barriers, safety ratings that are often inaccurate, and privacy issues. "Friend in Threat", however, smashes these inconveniences through mechanisms like voice recognition via NLP, AI distress detection, offline SMS alerts, and real-time emotion analysis, thus providing a much more advanced proactive security solution.

4. Working of System

The system under proposal is a voice-controlled women's safety app intended to identify distress using pre-defined keywords and AI-emotion analysis. It has a well-defined workflow from the time of launch to the activation of emergency alert. The following is a detailed description of each step in the working process.

4.1 App Launching Mode:

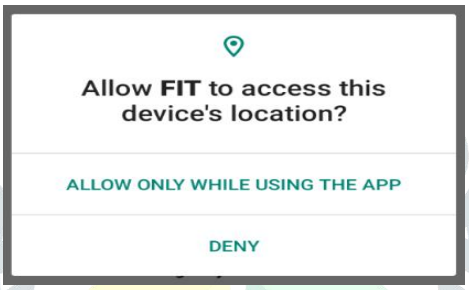
When the user opens the "FIT" application, the app gets initialized and checks if it is being opened for the first time or if it has already been configured. If the application has been installed recently, it continues with the configuration process, requesting permissions and registration of the user. If the application has already been used before, it simply goes to the main dashboard, where the basic functionalities are available. The launching mode makes the application ready to work correctly without needing repeated setup unless needed.

4.2 Checking GPS Status (ON/OFF):

The moment the app is opened, it automatically verifies if the GPS service is active on the device. The GPS is used to retrieve the user's actual location, which will be part of the emergency alert message. If the GPS is disabled, the application encourages the user to activate it. In the absence of GPS, the app cannot give a correct location when in a crisis, and hence it is an essential part of the system. The app checks the status of GPS consistently to ensure that it's available whenever there is an emergency.

4.3 Asking Permission for Location, SMS, and Sound Access:

The application asks the user for necessary permissions, such as access to location services, SMS, and microphone. The location permission is needed to fetch and send live coordinates in emergency situations. The SMS permission helps the application to send distress messages to the previously selected emergency numbers. Microphone permission helps the application to pick up specific voice commands that denote distress. If any of these are not granted, the app will not work correctly, and the user is informed of the necessity of granting them. In case a user denies any permission by mistake, the user is instructed on how to allow them from device settings.



4.4 Login Using Phone Number:

Users must log in through their mobile phone for security and personalization purposes. An OTP is sent to the phone number entered by the user for verification. When the OTP has been entered correctly, authenticated users can access the application. When the OTP has been entered incorrectly, the login will fail and prompt the user for another attempt. This way, only allowed authorized persons can use the app and achieve a bar against unauthorized access.

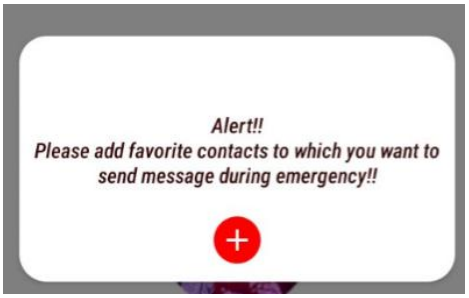
4.5 If Valid, Then Proceed to Signup:

The user should continue with the signup process once the OTP validation is done and effective. The application here fetches vital personal data such as the user's name, father's name, grandparent's name, village name, and pincode. All of this data is securely stored in the Firebase database and then used in the emergency alert message. The registration procedure is needed so that, in case of an emergency, the er contacts receive thorough information of the user. The app compels the user to provide all necessary information if any of the compulsory fields is left empty.

4.6 Checking for Pre-Saved Emergency Contacts:

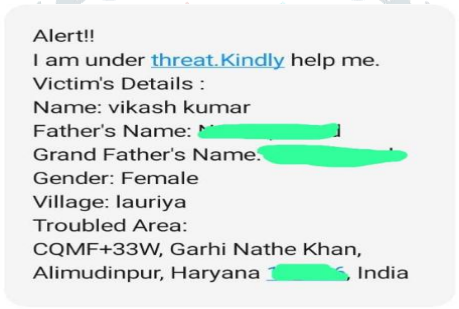
The app checks if the user has already included emergency contacts after the registration is completed. If the contacts list is empty, an alert box pops up asking the user to provide at least one emergency contact. This step ensures that the application has an honest receiver to deliver emergency alerts.

Several emergency contacts could be added by the user including local authorities, friends, or family members. Unless one contact is kept, the app will not let the user go any lower. This measure guarantees the application stays valuable in real-world emergency cases.



4.7 Voice Detection, Keyword Matching, and Emergency Alert Trigger*:*

The app is always ready to listen for certain call-out words which would indicate the user is in distress. After the user speaks, the app records the user voice and uses the Google Speech-to-Text API to both record and convert users speech into text. This text will then be processed with Natural Language Processing (NLP) functions to check whether any of the predefined distress keywords like, "Hey Ram", "Help me", or "Save me", are present. To further increase accuracy, the app applies the Levenshtein Distance Algorithm to allow for variations of the spoken words. If the user speaks a distress call which matches or is close to the defined keyword categories - an emergency alert is sent. The alert message is automatically generated to include the user email, current GPS location, associated father name, grand father , village name, and pincode, and SMS to the pre-selected emergency contacts. Note that if the device is connected to the Internet, then the alert can be sent using WhatsApp or Firebase notifications in addition to the SMS alert. The app ensures that the emergency contacts receive complete and appropriate information necessary to respond to the alert.



5. Methodology & System Architecture

To create the proposed system, a structured process was adopted, utilizing Android Studio as the major development environment and applying Kotlin, XML, MVVM architecture, Room database, and Firebase. Below is an overview of the methodology and setup employed:

5.1 Development Environment and Tools

Assess various technology stacks, frameworks, and development platforms to decide on the most appropriate ones for creating the Women Safety App.Android Studio Framework is a robust IDE for Android development that provides debugging, testing, and emulation capabilities. Android Studio uses a Gradle-based build system, Android Emulator, code templates and GitHub integration. Android Studio uses an Apply Changes feature to push code and resource changes to a running application. A code editor assists the developer with writing code and offering code completion, refraction and analysis. Applications built in Android Studio are then compiled into the APK format for submission to the Google Play Store.

5.2 Development Language Selection

Kotlin is a new language that has been developed to bridge the gap in Java while keeping it fully interoperable. Kotlin is Android's official language of development and allows for more concise, expressive, and safe development of applications relative to Java. Kotlin's one of the greatest strengths relative to other programming languages is that it is very concise. Kotlin handles higher-order

functions, lambda expressions, and inline functions easily and makes functional programming natural and more efficient. offer a more effective means of dealing with asynchronous programming.

Kotlin provides a more expressive, modern, and safer programming experience than Java. Its compact syntax, null safety, functional programming support, and coroutines make it a favorite among Android developers. Although Java is widely in use, Kotlin's productivity, maintainability, and performance advantages make it the superior choice for Android app development in the coming age.

5.3 Interactive UI Design

XML is being used for UI design in Android Studio as it provides a declarative, structured, and platform-independent way to define layouts. It allows separation of concerns, improves maintainability, synergizes with Android's resource system, and improves performance. Although Jetpack Compose (a new UI library) is gaining traction by making UI design available in Kotlin, XML is widely used because, for big projects, traditional layouts and view-based UI design are required.

5.4 Architecture Used

MVVM architecture is used to build the proposed system because it gives a cleaner architecture, improved separation of concerns, increased maintainability, enhanced testability, lifecycle-awareness, and optimized data handling. With reduced tight coupling between UI and business logic, applications become more scalable, stable, and manageable. As new Android development goes towards Jetpack components and coroutines in Kotlin, MVVM has become the go-to architectural pattern to design high-quality applications.

5.5 Voice Recognition and SMS Activation

Voice recognition and SMS activation in proposed system serve as features that offer increased safety for the user via hands-free emergency communication. These features make sure that even when the user cannot take manual control-on-demand action to alert and request for help, they would be in a position to do so via voice commands. Voice recognition in the app itself is done via the Google Speech-to-Text API, enabling the initiations of emergency functions via set voice commands. The application will then process audio input when the user says an emergency command like "Help me" or "Emergency": converting the audio to text and checking the text against the stored trigger phrase. Upon a positive match, the application will automatically trigger one such set emergency act, like alarm notification, location information sharing, or SMS activation. For greater reliability of service, this can be complemented by using background voice detection services that could listen for certain keywords, even when the app is running in the background. The other equally important part is SMS activation that inherently sends emergency messages to preselected contacts whenever the activation for such distress signals gets activated. SMS activation does this by using Android's SMS Manager API. Then in instances of emergency- for example, when voice detection picked up a distress word-the application will send out a distress SMS carrying the vital info about the user, including his GPS location and information together with the postal address.

5.6 Data Management

Room Database and Firebase Realtime Database have an impact on data consistency, reliability, and smooth syncing between local storage and cloud-based data management. The teamwork of these two database solutions makes the application run even when the internet is shaky or out of service. Room Database part of Android Jetpack's Persistence Library, helps users to organize and access local data storage on their device. Room databases offer a layer on top of SQLite to query, index, and fetch data while using the DAO (Data Access Object) interfaces for basic database tasks. Firebase Realtime Database, in contrast, serves as a cloud-based sync layer to keep information up-to-date across devices in real time. Firebase is a NoSQL database, which means it stores data than relational databases, and as a result, it updates the data for all users linked to the app.

5.7 Testing and Optimization

The app's testing and optimization involved several evaluation stages to check its reliability, efficiency, and user-friendliness. The development team ran unit and integration tests to confirm the app worked . Unit tests looked at individual parts the View Model and Model layers, to make sure data flowed and business logic was sound. The team also did integration tests to check if key systems like Room Database, Firebase, and SMS features worked well together. Besides checking functions, they tested performance in real-life

situations to see things like battery use how well voice recognition worked, and if SMS messages were sent . This helped them find ways to make the app work better without slowing it down.

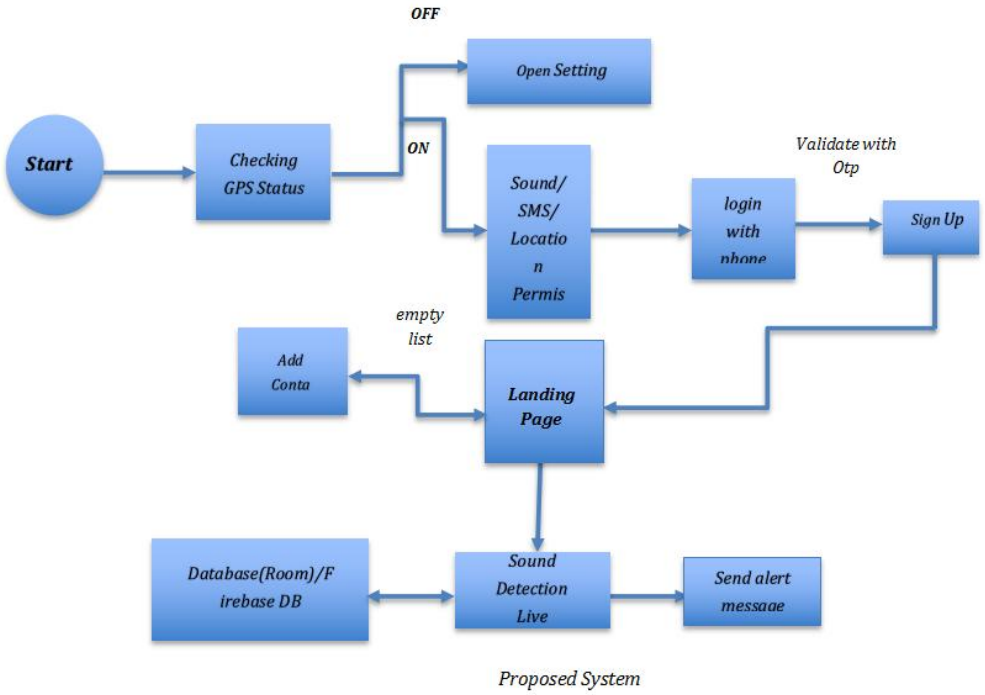
5.8 Security Considerations

The app was built with a strong focus on keeping user data safe while making sure it works well. We made data private by using special coding called encryption inside the Room Database. This helps stop anyone who isn't supposed to see the data from getting in. Important details like emergency contacts and other user information stay safe this way, even if someone tries to hack the device. On top of data safety, we carefully followed the rules from Android for handling permissions. The app asks for permission before it uses things like text messages, location, and the microphone. This makes sure users know how their data is used and gives them more control over what they want to share.

5.9 Deployment, Monitoring, Evaluation and Roll-out

The definition of deployment and roll-out focused on having the app ready for public and operational use while keeping the app operational and secure keeping in mind a large group of users. Part of the deployment plan was the thorough evaluation of the server infrastructure including the considerations for cloud storage for user data, relational database designs that could scale with the anticipated user base, and performance monitoring tools that would track the usage of the app and allow the developers to track any issues after production six months later. As part of the deployment plan, there was a requirement to monitor what app store delivery platforms would require in the deployment plan including Google Play Store factors such as what the delivery specifications were which included privacy policies, and security procedures amongst other considerations. Following the above methodology allowed for the established app will provide a reliable, responsive and safe emergency response for women utilizing planned architecture and compliant development processes.

6. Flow Diagram of Proposed System



Proposed System

7. Algorithms

7.1 Knuth-Morris-Pratt (KMP) algorithm:

The Knuth-Morris-Pratt (KMP) Algorithm provides optimal efficiency in pattern searching as its method for locating the substring in text is to minimize comparisons (by not using repeated comparisons). The KMP Algorithm preprocessess the pattern using the LPS (Longest Prefix Suffix) array, which creates the potential to shift in a smart fashion to the pattern next position when encountered with mismatches.

The KMP Algorithm consists of two steps, preprocessing and searching. The preprocessing step involves generating the LPS array, which lists and defines prefix longest matching suffix for every postion of the pattern. The searching phase involved moving the pattern and text pointer when a matching character is found. Pattern shifting at mismatches allows LPS values to be used to minimize comparisons instead of starting something new.

7.2 Advanced Encryption Standard (AES) Algorithm:

The Advanced Encryption Standard (AES) algorithm is at the heart of modern cryptography-strong, efficient and widely used. That's because it replaced the aging DES algorithm. AES processes data in 128-bit blocks, using keys that come in three sizes: 128, 192 or 256 bits.

AES works through multiple rounds. Each round is made up of four operations: the substitution of bytes for their equivalents, shifting rows into position, mixing columns together and adding a round key (via XOR). The number of rounds depends on the size of that key: 10 rounds for 128-bit keys, 12 for 192-bit keys and 14 for 256-bit keys.

7.3 Emergency SMS Dispatch Algorithm:

The SMS is composed by joining the user info with the current GPS coordinates. Android makes use of the SMS Manager API which uses a Broadcast Receiver pattern to send the message to user saved contacts. Once the application detects the keyword it extracts user info from the Room database in combination with location data to construct an SMS. The retry of an SMS in the case of failure is also facilitated through the application, which will retry to send the message when the network is back connectivity. It is common to implement with a retry algorithm, it cannot be guaranteed that the message will not be executed. A retry algorithm is commonly just a simple exponential back-off mechanism – waiting longer to retry resending to consume less resource.

7.4 Data Storage and Retrieval Algorithm:

Room database is an abstraction layer over SQLite. User data, such as contact information, personal data, etc., is stored locally and securely on the device using Room database. Room CRUD Operations (Create, Read, Update, Delete) are used to perform CRUD tasks on the data. Room uses features such as indexing and caching to enhance the CRUD operation for easy retrieval of frequently needed information. Firebase is used to store data remotely and retrieve data, providing the user's information in cloud data storage. Firebase enables the real-time synchronization of data with negligible latency.

Firebase implements a NoSQL system and synchronizes the data with the help of a JSON tree. It performs its synchronization using a Publish-Subscribe (Pub-Sub) model. The app publishes subscriptions to data nodes to specify the data node the app will be using and receive the updated information in real-time.

7.5 Location Tracking and GPS Algorithm:

The Global Positioning System (GPS) is a trilateration system that determines location by calculating distances from a minimum of three nearby satellites to the user's location. The concept of trilateration is consistent with the app, which uses GPS information to calculate the user's current location in latitude and longitude typically based on updated information as the device moves and as cell towers come in range. To significantly increase the accuracy of the location, Android uses GPS, Wi-Fi, and cell signals by using a weighted averaging algorithm that selects the better signal based upon signal strength and usability.

7.6Fuzzy Keyword Matching Algorithm:

The Levenshtein Distance Algorithm is used to accommodate minor pronunciation errors. Converts spoke input to text using Google Speech-to-Text API. The input is compared with pre-defined distress words ('Help me', 'Hey Ram', 'Save me'). if the distance is ≤ threshold (for example, 2character distance) → Trigger SOS.

7.7 An AI Model for Emotion Detection:

In order to train an AI model used for being able to identify or detect emotions, labeled datasets (data that has emotion data) are obtained, audio is preprocessed which involves reducing unnecessary noise and breaking it up into smaller segments, then the resultant audio files are utilized to extract features (the variables of human speech such as pitch, tone, placement, etc.,) using either utterances based on the MFCCs or a generalized spectrogram analysis. From there, machine learning models can like CNN (convolutional neural networks) trained on the spectrograms or LSTMs with the temporal data (long-short term memory) can process this feature extraction to classify emotions such as fear or distress. Lastly, the model must be trained, optionally retrained, or optimized, to be validated and verified before it is finally integrated into your final application. Conversing with someone while detecting emotions from human voice modulations in a non-threatening manner can be an ideal way to support emergency detection to create more accurate and responsive alerts in these types of situations.

8. Future Enhancements

8.1 Offline AI Capabilities:

To ensure continued smooth performance, the app will implement on-device machine learning models for speech recognition and distress detection. This will enable the app to still function without a constant internet connection, to allow the emergency alerts to be sent via SMS in cell reception 'dead-spot' environments as well.

8.2 Machine Learning-Based Anomaly Detection:

The app will contain anomaly detection algorithms which will be able to analyze the user's regular movements and behavior patterns. If the user's actions deviate from predictable patterns, such as sudden stops in an isolated area, unusual delays, or repeat distress signals, an automatic alert will be generated so that threats can be detected before the user realizes they've even been threatened.

8.3 Emotion and Sentiment Detection:

The system will employ AI driven emotion and sentiment analysis which identifies stress, fear, or panic in the voice of the user. The system will employ Natural Language Processing (NLP) and Deep Learning models to analyze the change in speech patterns, tone and intensity to identify if the user is distressed or not. The models will be capable of indicating distress even when the user doesn't employ pre-defined keywords.

8.4 Integration with Smart Wearables and IoT Devices:

Future updates will include integration with smartwatches, Bluetooth-enabled safety devices, and IoT-based security gadgets. Users will be able to trigger emergency alerts with simple gestures, such as pressing a smartwatch button, shaking the phone, or using a voice command through connected devices.

9. Conclusion

This research depicts the potential of both voice recognition via NLP and AI to propel those applications that are concerned with women's safety in massive ways forward. While other women's safety applications other than 'Friend in Threat' do not utilize fuzzy keyword detection, emergency alerting in real-time, and an in-development offline model, 'Friend in Threat' uses them to achieve an even greater milestone in usability and accessibility in action. Advances in speech analysis deep learning may provide greater resiliency for the real-world applications in emergency scenarios as well. Apps that make safety for women more effective provide base features such as GPS tracking, pre-set emergency contacts, and standard SOS alerts with degree of overall features.

None of these currently hold promise for AI-based automation, fuzzy keyword detection, sentiment analysis, predictive geofencing and a lot of other avenues of automation. Several WHO efforts have listed areas which AI can enhance better responses in emergencies and will go on to suggest experts in AI encompass three elements of monitoring social behavior: (1) speech recognition, (b) behavioral anomalies, and (c) real-time distress. The paper offers much more APIs to conventional safety apps but adds the extra AI boost with voice recognition, fuzzy keyword matching, emotion detection, and offline notifications as other "conventional" apps would qualify as "hard" requirements as an "intelligent, proactive solution" to solve women's safety and security.

References:

[1] Manazir, S. H., Govind, M., & Hassan, R. (2019). My Safetipin Mobile Phone Application: Case Study of E-Participation Platform for Women Safety in India. Journal of Scientometric Research, 8(1), 47-53

[2] A. Kumar and R. Yadav, "Analysis of women's safety apps and their effectiveness: A case study on India," J. Safety Res.*, vol. 45, no. 2, pp. 115-130, 2021.*

[3]. Kumar, Ashish, and Rajesh Yadav. "Analysis of Women's Safety Apps and Their Effectiveness: A Case Study on India." Journal of Safety Research *45, no. 2 (2021): 115-130.*

[4]. Patil, S., Rajput, P., Patil, J., Yemul, M., & Patankar, A. J. (2021). GoSafe: A Safety and Real-Time Emergency Mediator System between Police Officials and Citizens of India. International Research Journal of Engineering and Technology (IRJET), 8(4), 1669-1674.

[5] World Health Organization, Violence against women prevalence estimates, 2018*, WHO, 2021.*

[6]. Thakare, Rajani T., and M. R. Shahade, "Review on Android Based SOS Emergency Alert Button," International Journal of Research in Science & Engineering, vol. 1, 2015.

[7]. Android Developers, Location APIs. URL: http://developer.android.com/google/playservices/location.html

[8]. bSafe. (2024). bSafe – Personal Safety App. https://www.getbsafe.com

[9]. "WOMEN'S SECURITY", Android App developed by App Soft India, December 17, 2019. https://play. google. com/store /apps/ details? id= com. Zayaninfotech. security& hl=en.

[10] I. S. A. E. U. Sudhakar, "A Mobile Based Women Safety Application (I Safe Apps)," IOSR Journal of Computer Engineering, vol. 17, no. 1, pp. 29-34, 2023. [Online]. Available: www.iosrjournals.org.

[11]. Monisha, D. G., Monisha, M., Pavithra, G., & Subhashini, R. (2016). Women Safety Device and Application-FEMME. Indian Journal of Science and Technology, 9(10). DOI: 10.17485/ijst/2016/v9i10/88898.

[12] "Existing Women Safety Apps Review," International Research Journal of Engineering and Technology (IRJET), vol. 10, no. 4, pp. 1514-1516, Apr. 2023. [Online]. Available: www.irjet.net

[13]. Hyndavi, V., N. Sai Nikhita, and S. Rakesh. "Smart wearable device for women safety using IoT." In 2020 5th International Conference on Communication and Electronics Systems (ICCES), pp. 459-463. IEEE, 2020

[14]. N. Ramesh Kannan , S. Sujitha, S. Ganapathy Subramanian, "Women Safety Mobile App," International Journal on Cybernetics & Informatics (IJCI) Vol. 10, No.1/2, May 2021.

[15]. Westmarland, N., Hardey, M., Bows, H., Branley, D., Chowdhury, M., Wheatley, K., & Wistow, R. (2013). Protecting Women's Safety? The use of smartphone 'apps' in relation to domestic and sexual violence. SASS Research Briefing No. 12, Durham University.

[16]. Juhitha, S., Pavithra, M., & Archana, E. (2020). Design and Implementation of Women Safety System Using Mobile Application in Real-Time Environment. International Journal of Research in Engineering, Science and Management (IJRESM), Volume: 3, Issue: 4, Pages: 21-26.

[17]. Kataria, K., Khade, R., Kurhade, R., Pende, A., & Chanderi, S. (2022). A Survey Paper on Android App for Women Safety. International Journal of Research Publication and Reviews, 3(11), 1905–1911.

[18]. Hossain, E., Rahman, W., Islam, M. T., & Hossain, M. S. (2019). Manifesting a mobile application on safety which ascertains women salus in Bangladesh. International Journal of Electrical and Computer Engineering (IJECE), 9(5), 3660–3667