International Journal of Advance Research in Science, Management, Engineering and Technology
Volume 1, Issue 5 | November - December 2026 | www.ijarsmet.com
ISSN: 3139-1141

A MULTIMODAL AI-ASSISTED WOMEN SAFETY PROTOTYPE USING RANDOM FOREST, OPENCV, SPEECH RECOGNITION AND GPS-BASED SOS ON STREAMLIT

Neha Keshav Mahale
Department of Computer Application
Zeal Institute of Business Administration, Computer Application & Research (ZIBACAR), Savitribai Phule Pune University, Pune, India
Email: mahaleneha16@gmail.com
ORCID: 0009-0003-0618-2183

***Abstract*—*This paper presents and critically evaluates a prototype women-safety application that integrates machine-learning-based risk prediction, computer-vision-based video analysis, voice emergency detection, GPSbased SOS messaging, and incident analytics in a single Streamlit dashboard. The prototype takes contextual safety information as input, including device type, panic-button status, response time, day, location category, age, and emergency-contact status, and passes it to a stored Random Forest classifier. Its computer-vision module uses OpenCV to measure inter-frame motion in uploaded video as an early-warning indicator, while a background listener detects predefined emergency phrases from microphone input. The supplied dataset contains 1,000 incident records; descriptive analysis shows a mean response time of 16.05 minutes, 504 panic-button incidents, and 482 assault records. The results demonstrate a functioning safety workflow, but the prototype cannot yet be considered a crime-detection system, because the project files contain no video annotations, training code, test metrics, or end-to-end threat classifier. The paper concludes with the validation, privacy, and deployment work required before field use.*

Keywords: Women Safety, Machine Learning, Computer Vision, OpenCV, Emergency Alert System, Multimodal Safety Framework

1. INTRODUCTION

Women-safety systems should facilitate rapid detection, communication, and response while minimising false positives and privacy invasions. Computer vision can enable this by detecting objects, persons, poses, or spatio-temporal activity in video. Prior work has used convolutional neural networks to encode spatial features and long short-term memory (LSTM) networks to encode temporal features for violence recognition, reporting the benefit of analysing sequences rather than individual frames [1]. One real-time violence-recognition approach combined human detection, weapon detection, convolutional neural network (CNN) features, LSTM temporal modelling, and activity classification [2].

The supplied project adopts a broader, multimodal approach. It implements a contextual risk model, video inter-frame motion analysis, voice emergency detection, location capture, and messaging as a cohesive prototype because a safety incident can have multiple contributing factors and may be indicated by several weak signals rather than one reliable descriptor.

A. PROBLEM STATEMENT

Traditional emergency applications mainly rely on user interaction initiated after the fact. A person may not be able to press the panic button, type a message, or provide precise location details. A CCTV camera may record an incident without anyone actively viewing the video stream. The problem the current project addresses is designing an integrated system that can: (1) estimate contextual risk, (2) detect interframe motion in the uploaded video, (3) identify predefined emergency phrases, (4) capture the location, and (5) message a trusted person.

The current project should be considered as a threat indication and emergency assistance system, not an autonomous crime-detection system. Its own motionanalysis logic only evaluates movement intensity, without determining whether a crime has occurred [3].

B. OBJECTIVES

1. Design a browser-based women-safety dashboard using Streamlit
2. Predict contextual risk using a stored machinelearning model
3. Analyse uploaded CCTV/security camera video for inter-frame motion
4. Detect predefined emergency phrases using background voice recognition
5. Capture current GPS coordinates and create maplinked messages
6. Implement panic-button and trusted-contact procedures
7. Visualise incident records using charts and descriptive statistics
8. Identify validation, privacy, and deployment gaps for the prototype [3]–[5]

ISSN: 3139-1141

C. SIGNIFICANCE OF THE STUDY

The study is significant for three reasons. First, it shows how weak signals from context, video, voice, and location can be combined into one assistance workflow rather than treated as separate tools. Second, it states plainly what a prototype of this kind can and cannot claim, which helps avoid overstating the capability of motion-based or rule-based safety tools. Third, it identifies the validation, privacy, and deployment requirements that must be met before such a system can be trialled with real users.

2. LITERATURE REVIEW

A. EXISTING SYSTEMS

The existing systems typically fall into manual SOS applications, CCTV cameras, panic-button wearable devices, and research systems that implement action or violence recognition. Manual SOS tools are straightforward but may be ineffective if the user is incapacitated or unable to interact with a mobile application. CCTV cameras may cover a wide area but require constant monitoring, and false positives may abound in challenging lighting or overcrowded conditions. Research systems have implemented You Only Look Once (YOLO)-based live surveillance, activity recognition, SOS gesture detection, location tracking, and automatic alerting [6]. Convolutional neural network–long short-term memory (CNN–LSTM) systems encode spatial and temporal information, and hybrid object and weapon detection systems identify violence [1], [2]. A deep-learningbased women-safety system has also combined live surrounding analysis, speech-related signals, GPS tracking, and SMS/email alerting [7]. Table I summarises the studies reviewed and how the present work relates to each.

TABLE I SUMMARY OF RELATED STUDIES

Ref.	Approach	Relation to present work
[1]	CNN spatial features with LSTM temporal modelling for violence recognition; evaluated on a 2,000-video dataset.	Focuses on video classification; contextual data, voice, and SOS messaging are outside its scope.
[2]	Human detection, weapon detection, and VGG19/LSTM processing for real-time violence recognition.	Principal focus is visionbased detection; contextual risk and messaging are not its focus.
[6]	YOLOv8-based live surveillance with tracking, abnormal-situation detection, SOS gestures, and alerting.	Needs a trained deeplearning detector; this prototype uses lightweight motion analysis and states its validation limits.
[7]	Deep-learning computer vision with live surrounding analysis, speech signals, GPS tracking, and SMS/email alerts.	Closest precedent for multimodal alerting; this work centres on contextual risk, incident analytics, and transparent limits.

B. RESEARCH GAP

The main limitation of the existing approaches is that detection is framed as a single-model task, and their evidence is tied to labelled video data. In reality, threat assessment also involves response time, panic-button status, location, trusted-contact notification, and the user's ability to communicate. The current project addresses these integration challenges, but its computer-vision component is only a prototype and not a trained violence-recognition model [3], [5]. The work therefore builds on the vision-based precedent [1], [2] by treating video as one weak signal among several, and contributes an integrated, openly documented prototype rather than a new detection model.

3. METHODOLOGY / PROPOSED SYSTEM

A. PROPOSED SYSTEM

The proposed system has five modules:

1. Contextual risk prediction: A stored model makes predictions about the risk category based on structured input data. The Streamlit interface only labels it as a Random Forest model [3].
2. Video movement analysis: The user uploads a short video, and OpenCV samples the frames at intervals, resizing them to 640 × 360 pixels, converting them to grayscale, blurring similar consecutive frames, and calculating the absolute inter-frame difference, which is thresholded to calculate the percentage of moving pixels [3].
3. Voice emergency detection: A background listener captures microphone input, performs speech recognition, normalises the resulting text, and evaluates it for predefined emergency phrases, such as "help," "I am in danger," "please help me," or "someone is attacking me" [4].
4. SOS and location: The browser obtains highaccuracy GPS coordinates, prepares a Google Maps link, SMS link, and WhatsApp link, and the user presses Send on their messaging client after reviewing the location [3].
5. Incident analytics: Charts help describe the case type, location category, device type, panic-button status, and emergency-contact status from a CSV file.

B. METHODOLOGY / ARCHITECTURE

The system architecture is organised into six layers, as shown in Fig. 1.

Input Layer: This layer takes in multimodal data from the user: HTML form entries, video recordings, microphone input, the manually pressed panic button, and real-time location.

Pre-processing Layer: This layer prepares the multimodal data for processing. Categorical data is formatted, video frames are sampled, resized, and

ISSN: 3139-1141

converted to grayscale, and audio is converted to text and normalised.

Analysis Layer: This layer analyses the processed data for possible threat indicators. It calculates prediction probabilities from the stored ML classifier, analyses motion within the video, and matches voice phrases. These outputs feed the decision layer.

Decision Layer: This layer combines the analytics to assign a severity level. Movement intensity is grouped into low, medium, and high levels, and a 30-second cooldown is enforced between voice-triggered alerts to limit repeated emergency messages.

Response Layer: This layer prepares the response to a detected or manually triggered emergency. It builds a location-specific message containing the GPS coordinates and a Google Maps link, and generates pre-filled SMS and WhatsApp links for the trusted contact. The user must press Send: the application does not send messages automatically, call the police, or contact emergency services.

Analytics Layer: This layer supports analysis after an incident. Event logs and situation data are stored in pandas dataframes and used to create charts and reports internally.

Because the motion level is only an alert indicator and not a validated threat score, the system is designed to be human-in-the-loop. Deploying it in production would require a trained temporal action classifier, confidence thresholds, time-stamped events, privacyprotected authentication, user consent, and fully tested emergency response.

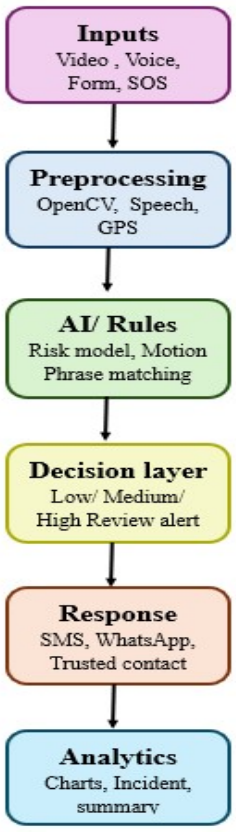


Fig. 1. System architecture showing the input, preprocessing, analysis, decision, response, and analytics layers. The architecture is reconstructed from the supplied application and listener code [3], [4].

C. TECHNOLOGIES / ALGORITHMS / MODELS USED

- Python: Application and data-processing language
- Streamlit: Dashboard and page navigation
- pandas and NumPy: Tabular dataframes [8]
- scikit-learn: Compatibility layer for the stored model • joblib: Model serialization
- OpenCV: Video decoding, resizing, grey scaling, blurring, frame differencing, thresholding, and pixel counting [3]
- Random Forest prediction: Used for the risk prediction; the supplied files do not contain model training code or the model itself, so its hyperparameters are unknown [3], [9]
- SpeechRecognition: Microphone input and Google speech recognition in the background listener [4]
- Rule-based phrase matching: Lowercasing, punctuation removal, repeating character reduction, and substring matching
- Browser Geolocation API: Latitude and longitude for SOS message location [3]

D. DATASET

The safety.csv file contains 1,000 records with the following variables: device type, GPS, panic-button status, incident type, response time, outcome, day, time, location category, age, and emergency-contact status [5]. These variables support several exploratory analyses: response times can be examined by location to identify high-risk zones, and panic-button use can be compared with incident outcomes to see how users interacted with their devices. Overall, the data provide a useful base for studying how emergency responses are handled and for improving alert triggers and contact workflows.

Table II summarises the descriptive analysis of the supplied file.

TABLE II DESCRIPTIVE STATISTICS OF THE SAFETY DATASET (N = 1,000)

Variable	Category	Count (%)
Device type	Panic Button Device	344 (34.4)
	Smartwatch	334 (33.4)
	Mobile	322 (32.2)
Panic button	Yes	504 (50.4)
	No	496 (49.6)
Incident type	Teasing	518 (51.8)
	Assault	482 (48.2)
Outcome	Resolved	344 (34.4)
	Unresolved	334 (33.4)
	In Progress	322 (32.2)

ISSN: 3139-1141

Location category	Moderate-risk Zone	346 (34.6)
	Safe Zone	338 (33.8)
	High-risk Zone	316 (31.6)
Emergency contact	Notified with Confirmation	388 (38.8)
	Notified with Delay	316 (31.6)
	Notified	296 (29.6)
Age (years)	Mean	30.41
Response time (min)	Mean	16.05

These are descriptive statistics of the supplied file, not prevalence rates of populations of interest. The file is a table and does not contain the video frames, bounding boxes, labels, and audio needed to evaluate a computer-vision-based violence classifier [5].

Thus, it can be used for dashboard analytics and structured risk modelling, but not for evaluating the prototype's visual threat detection. The distributions of incident type, location category, and response outcome are shown in Fig. 2.

Project scripts also describe the file preparation. The first script identified 500 duplicate rows, with the first 500 identical to the last 500. After deduplication, a 500-row file remained, with 259 teasing and 241 assault cases [5]. Another machine-learning file adds a risk-level column with 337 medium, 86 low, and 77 high records, indicating a class imbalance. The statistics described above apply to the 1,000-row file and should be interpreted with this caveat.

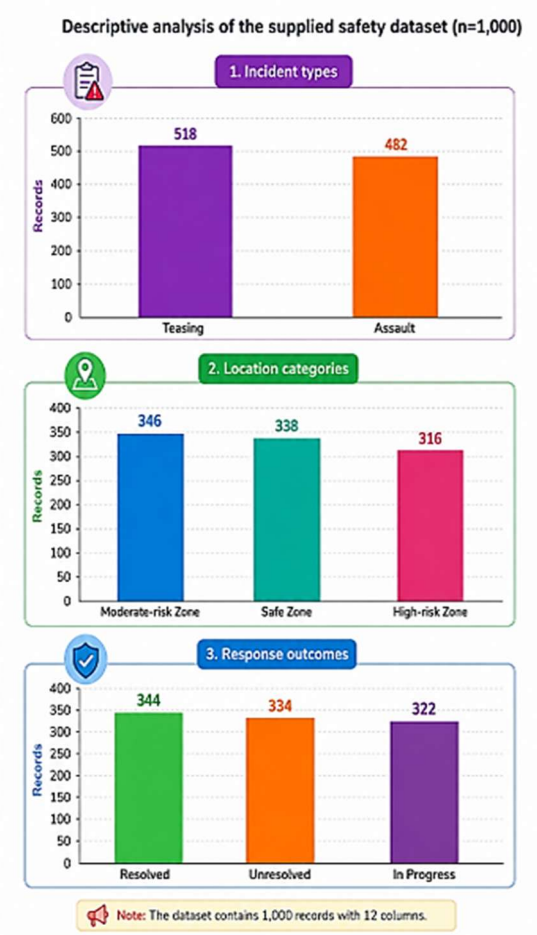


Fig. 2. Descriptive distribution of incident type, location category, and response outcome in the supplied 1,000-record dataset [5].

E. SYSTEM IMPLEMENTATION AND EXPERIMENTAL SETUP

The application is organised as a multi-page Streamlit dashboard. At startup, it attempts to load the serialised model from models/risk_prediction_model.pkl. If it fails, the risk prediction page announces that the model is missing [3]. The user selects the device type, panic-button, response time, day, location category, age, and emergency-contact status. The application builds a one-row pandas dataframe, invokes predict and predict_proba, and displays the predicted class and probabilities [3].

For the video analysis, the user uploads an MP4, AVI, MOV, MKV, or WEBM video, and the sampling interval and motion threshold can be adjusted. Each sampled frame is resized to 640 × 360 pixels; consecutive frames are desaturated and blurred, and their absolute inter-frame difference is thresholded to calculate the percentage of moving pixels [3]. For the voice recognition, the listener adjusts to ambient noise levels, constantly captures short audio snippets, performs speech recognition in the en-IN language, normalises the resulting text, and evaluates it for emergency phrases with a 30-second cooldown between alerts [4].

Apart from phrase matching, the Voice Safety page also accepts a short voice message, converts it to text, and scores it with a basic natural language classifier. The classifier uses TF-IDF features with unigrams and bigrams and a class-balanced logistic regression model, trained on 16 everyday sentences and 24 distress sentences. Its probability is combined with the keyword matching, and scores of 70 and above are labelled high risk, while scores of 40 and above are medium risk [3]. Because the training data is limited and handcrafted, this is only a proof-of-concept demonstration of the voice safety workflow.

The application also has supporting modules that facilitate the prototype as a fully functioning service. Users can register and log in with hashed passwords saved in a local JSON file. Other pages contain an emergency alert screen, a live location with nearby hospitals and police stations sorted by Haversine distance, a safety timer, trip and check-in planning, helpline numbers, incident reporting, and a keywordbased assistant that responds in English, Hindi, and Marathi. The nearby-place list is only an example and needs to be connected to a reliable map or directory service in production [3]. In the movement page, the pixel difference threshold is set to 20, and the movement score of more than 60 triggers a highmovement alert [3].

AI risk prediction (Random Forest): The risk prediction page implements an AI assistant that

ISSN: 3139-1141

estimates how risky a situation is, so the user knows what precautions to take before an emergency occurs. The user provides seven input variables about the situation: device type, panic-button activation, response time, day, location category, age, and emergency-contact status. The model only uses these variables, and the page stores the selected values as a one-row table and passes it to the stored Random Forest classifier, an ensemble of decision trees that combine the individual tree predictions into one class prediction. It returns the predicted class along with class probabilities and displays them as low, medium, or high risk [3]. A medium or high prediction encourages the user to take action, such as reviewing the location, notifying a trusted contact, setting off the safety timer, or visiting the emergency alert page. A low prediction should not suggest complacency because it is only an estimate based on the input variables. Within the application, it is the only component that implements machine learning for structured data, and its limitations are described in the Limitations section.

Voice safety analysis: The Voice Safety page is useful when the user cannot type but can speak. It has four steps: (1) the user records a short voice message, (2) the recorded message is converted to text by the speech recognition module, (3) the text is normalised and scanned for distress or unsafe-situation indicators using the TF-IDF and logistic regression classifier and keyword matching described in the preceding section, and (4) the results are displayed to the user as a safety score and estimated risk level [3]. The analysis is performed on the text content only; the implementation does not contain a deep-learning voice-emotion recognition model that could interpret tone, pitch, or emotional cues. It supplements the always-listening phrase matcher, which is designed for hands-free emergency use, and works with the other modules to give the user an alternative way to express concern. Because it relies on predefined indicators, a small set of training examples, and the accuracy of the speech-to-text recogniser, it may fail to identify indirect expressions of distress, multilingual input, accents, or background noise.

Movement detection: The Movement Detection page applies the video analysis pipeline to the useruploaded video. It warns the user about potential safety concerns by sampling the video at intervals, resizing the frames, grey-scaling them, blurring similar consecutive frames, thresholding their absolute inter-frame difference, and reporting the percentage of moving pixels as an indication of movement. It displays the mean and peak movement values, a 0–100 movement score, and a low, medium, or high movement level and warns the user if the movement is high [3]. This information can help the user focus on reviewing the video content around the time when significant movement was detected. High movement does not always indicate criminal activity, as running, dancing, or camera movements can also register as motion. Thus, it is only a proof-of-concept movement detector that requires human eyes to identify potential threats.

Emergency alert (SOS) and trusted-contact workflow: The Emergency Alert page provides the user with a fast way to send an SOS. The user enters a trusted emergency-contact name and mobile number, and this contact becomes the recipient of emergency messages. When the user triggers the SOS feature, the application retrieves the current location, prepares an emergency message with the coordinates and a Google Maps link, and generates SMS and WhatsApp links addressed to the emergency contact [3]. The links open the default messaging applications with the message pre-filled, and the user has to send the message themselves. The application does not send the message, call the police, or notify others on the user's behalf and cannot guarantee that the message will be delivered or received. The SOS feature is only a convenience tool to reduce the time and effort spent on addressing an emergency.

Live GPS location: The Live Location page uses the browser's Geolocation API to obtain high-accuracy latitude and longitude after the user grants permission. It displays the current lat-long and shows a map link that the user can share [3]. It also lists nearby hospitals and police stations sorted by distance using the Haversine formula, but these are only examples. The same location information is used to prepare the emergency message content in the Emergency Alert page, so the emergency contact also sees the user's current lat-long. In the web version, the location is only fetched when the user initiates an action, rather than being tracked constantly in background, for privacy reasons. The location is sensitive personal information, and thus, its collection and processing should be limited to cases when the user explicitly authorises it.

Safety timer and future risk assessment: The safety timer helps the user set a precautionary reminder before travelling somewhere or doing something unsafe, such as a 30-minute safety session before going to a railway station. It acts as a gentle reminder to check in and confirm that the user is safe within the specified timeframe. The trip planning and check-in page provide a similar function but for future situations. The user describes a planned scenario, such as walking through a railway station at night, and receives an estimated risk level and probability so they can decide whether to change plans. This feature differs from the current risk prediction because it is a forward-looking assessment rather than an evaluation of the current circumstances. The risk score is generated using the Random Forest model, but it does not describe this as a separate training process or a standalone predictive model.

ISSN: 3139-1141

AI safety assistant: The assistant accepts user input in the form of a question and matches it against predefined keywords and intents. When there is a match, it provides a prepared response, and the responses are available in English, Hindi, and Marathi [3]. The supported intents include emergency guidance, SOS usage, harassment guidance, travelling alone, live location, helpline, voice safety, movement detection, and risk prediction. The assistant is essentially a rule-based, keyword-triggered, AIpowered chatbot. It is not a general-purpose large language model and is limited to the predefined intents and responses. Its value lies in providing access to the other modules' functions in a conversational manner, which can be especially useful for users who prefer to type in regional languages.

Incident reporting and safety information modules: The Report Incident page asks the user to choose an incident type: harassment, suspicious activity, unsafe location, or other, and to provide a description before submitting the report [3]. The ability to record such incidents promotes documentation and awareness of unsafe places and behaviour. A submitted report should not be considered a formal complaint to the police. Three supporting information modules accompany the predictive and emergency features: the Safety Tips section provides preventive suggestions on personal safety and potentially unsafe situations, the Helpline Numbers section provides quick access to emergency and support contacts in a limited time, and the Safety News section provides safety-related information to keep users up to date. These modules are informational and have no quantitative evaluation.

Classification of components: The Random Forest risk model and TF-IDF logistic-regression voice recognizer are the machine-learning components; the phrase listener is rule-based; movement detection is a computer-vision prototype based on frame differencing; and SOS, live-location, timer, helpline, tips, news, and reporting pages are utility and support modules.

4. RESULTS AND DISCUSSION

A. RESULTS

The available results are prototype-level and descriptive, not a formal model evaluation. The supplied data contain 1,000 records with a mean response time of 16.05 minutes. Assault and teasing records occur in similar proportions, and panic-button activation is nearly balanced between Yes and No records. Mean response time differs only slightly across devices (Table III) [5].

TABLE III MEAN RESPONSE TIME BY DEVICE TYPE

Device type	Records	Mean response time (min)
Smartwatch	334	16.51
Mobile	322	16.38
Panic Button Device	344	15.30

Location and response outcomes show a descriptive pattern (Table IV): resolved outcomes account for about 40.2% of Safe Zone records, versus 29.7% in High-risk Zone and 32.9% in Moderate-risk Zone records [5]. These associations are exploratory; the dataset does not establish causality and does not document how the records were generated or sampled.

TABLE IV RESOLVED OUTCOMES BY LOCATION CATEGORY

Location category	Records	Resolved (%)
Safe Zone	338	40.2
Moderate-risk Zone	346	32.9
High-risk Zone	316	29.7

No accuracy, precision, recall, F1-score, ROC-AUC, confusion matrix, latency benchmark, or video-level detection score can be reported from the supplied project files. The risk model is provided as a pretrained artefact: its training labels, data split, feature encoding, and test set are not included. Similarly, the motion module reports movement intensity, not truepositive or false-positive threat detection. This distinction is crucial: published CNN–LSTM violence-recognition work reports classification performance only after training and verification on labelled video datasets [1], whereas the supplied project does not provide equivalent evaluation evidence.

The additional modules described in the System Implementation section have been described functionally and have not been evaluated quantitatively. No accuracy, response-time, detectionrate, or user-satisfaction figures are reported for the risk-prediction, voice, movement, SOS, location, timer, assistant, or reporting modules as the supplied files contain no such measurements.

B. ADVANTAGES

1. **Multimodal support:** contextual data, video motion, voice phrases, GPS, and manual SOS are combined [3], [4].
2. **Simple deployment model:** Streamlit provides a fast interface for demonstrations and pilot testing [3].
3. **Human-in-the-loop design:** motion results prompt review rather than assuming every movement is criminal [3].
4. **Actionable response path:** the system prepares a location-linked message for a trusted person [3].
5. **Configurable video processing:** frame sampling and motion sensitivity can be adjusted for speed and scene conditions [3].
6. **Historical analytics:** incident records can be explored by device, incident type, location, outcome, and contact status [5].

ISSN: 3139-1141

7. **Extensible architecture:** the motion heuristic could be replaced with a trained action-recognition model.

C. LIMITATIONS

1. **No validated computer-vision threat model:** frame differencing measures motion with no capacity to distinguish assault from dancing, running, camera shake, crowds, or lighting changes.
2. **No supplied training pipeline:** the stored Random Forest model cannot be audited for leakage, calibration, fairness, or generalisation.
3. **No labelled vision dataset:** the CSV is tabular incident data, not a video benchmark.
4. **Rule-based speech detection:** exact or near-exact phrases might miss indirect, multilingual, noisy, or non-verbal distress; speech recognition is also dependent on network and microphone conditions.
5. **Manual message sending:** the browser opens SMS/WhatsApp links but the user must press Send [3].
6. **Privacy and security exposure:** GPS, audio, and video are sensitive data and require consent, encryption, access control, retention limits, and audit logs.
7. **Potential bias:** a model trained on narrow locations, devices, ages, accents, lighting conditions, or incident definitions may perform unevenly across groups.
8. **Dataset validity is unknown:** the supplied CSV does not describe collection protocol, ground-truth rules, missingness, or whether records are real, simulated, or synthetically generated.
9. **No operational response guarantee:** an alert interface is not equivalent to a verified police, ambulance, or security response channel.

5. CONCLUSION AND FUTURE SCOPE

A. CONCLUSION

The project provides a useful prototype for an AIassisted women-safety platform. Its strongest contribution is system integration: a Streamlit dashboard connects contextual risk prediction, OpenCV motion analysis, voice-triggered emergency detection, GPS-based SOS preparation, and incident analytics [3], [4]. The supplied 1,000-record dataset supports descriptive reporting but does not validate computer-vision threat detection [5]. The current vision module should therefore be described accurately as motion-based early warning [3]. A credible research-grade system requires labelled video data, a reproducible training pipeline, independent testing, calibrated probabilities, privacy safeguards, and evaluation of false alarms and response outcomes to evolve the prototype from a demonstration interface into a rigorously evaluated multimodal safetyassistance system.

In addition to these core components, the prototype has a user-assisted SOS and trusted-contact workflow, on-request GPS location, safety timer and plannedsituation assessment, rule-based multilingual safety assistant, incident reporting, and safety information pages. These modules make the system a more complete safety-support service but are still userassisted and not substitutes for verified emergency response.

B. FUTURE SCOPE

Future work should replace frame differencing with a benchmarked temporal model such as a 3D CNN, CNN–LSTM, transformer-based video encoder, or pose-and-object fusion model; CNN–LSTM and hybrid detection systems provide relevant precedent [1], [2]. A suitable dataset should include consented, de-identified clips with action labels, scene context, lighting conditions, camera motion, and difficult nonthreat examples. Model evaluation should use person- and scene-independent train/test splits and report precision, recall, F1-score, false-alarm rate, detection latency, and calibration.

The system could also add person detection, weapon detection, pose estimation, fall detection, crowddensity estimation, geofenced risk maps, multilingual speech recognition, offline wake-word processing, and sensor fusion with smartwatch accelerometers; person, weapon, and temporal activity detection are already used together in related surveillance research [2]. These additions should be tested for incremental value rather than added without ablation studies.

Deployment research should focus on privacypreserving processing, on-device inference where possible, encrypted event storage, consent and deletion controls, interpretable alerts, and escalation policies that keep a human operator involved. Privacy and ethical implications of video surveillance are central deployment considerations. Field trials should measure not only model accuracy but also time to acknowledgement, false-alarm burden, user trust, accessibility, and outcomes after alerts.

REFERENCES

[1] A. Dilshad and K. Agrawal, “Violence detection using convolution neural network and long short-term memory,” in *Proc. Int. Colloq. Comput., Commun., Control, Manage. (ICCCCM)*, 2024, doi: 10.1109/ICCCCM61016.2024.11039900.

[2] N. Jain, V. Gupta, U. Tariq, and D. Hemanth, “Fast violence recognition in video surveillance by integrating object detection and Conv-LSTM,” *Int. J. Artif. Intell. Tools*, 2022, doi: 10.1142/s0218213023400183.

[3] Women-Safety Prototype Project, “app.py: Streamlit application source code,” unpublished project file.

ISSN: 3139-1141

[4] Women-Safety Prototype Project, "background_listener.py: Background voiceemergency listener source code," unpublished project file.

[5] Women-Safety Prototype Project, "safety.csv: Incident dataset (1,000 records)," unpublished project file.

[6] D. Aruna Kumari *et al.*, "Women safety and surveillance system using YOLOv8," 2025.

[7] A. Anju *et al.*, "Third vision for safeguarding women with a live surrounding update using deep-learning for computer vision," 2021.

[8] Women-Safety Prototype Project, "requirements.txt: Python dependency list (Streamlit, pandas, NumPy, scikit-learn, matplotlib, seaborn)," unpublished project file.

[9] Women-Safety Prototype Project, "predict_risk.py: Risk-prediction script," unpublished project file.