International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 12 | Dec – 2025 DOI: 10.55041/ISJEM03319
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

A Real-Time Mobile Framework for Blackmail Detection and Emergency Response

Amirthavarshini B[1], Mr. Perumal P[2], Anu Deepthi V[3], Brinda Iswarya Lakshmi R[4]

[2]Professor, Department of Computer Science and Engineering, Sri Ramakrishna Engineering College, Coimbatore, Tamil Nadu, India

[1,3,4]Student, Department of Computer Science and Engineering, Sri Ramakrishna Engineering College, Coimbatore, Tamil Nadu, India

I. ABSTRACT

This paper presents a mobile-centric blackmail detection system powered by transformer-based language modeling. The architecture employs tokenized SMS parsing, zero-temperature autoregressive decoding, and structured prompt engineering to classify textual threats using probabilistic output constraints. Real-time inference is integrated through asynchronous RESTful communication with a hosted NLP engine. Threat-level predictions are stored in persistent key-value structures, while threshold breaches trigger multithreaded alert workflows, geolocation-driven intent dispatching, and buffered CSV report generation. Results confirm that integrating large language models with mobile runtime handlers enables efficient, deterministic blackmail detection and context-aware response using on-device asynchronous operations and permission-managed system APIs.

Keywords:- Blackmail Detection, Cyber Security, Artificial Intelligence, Emergency Response, SMS Analysis, GPS Analysis

II. INTRODUCTION

With the exponential rise in mobile communication, SMS remains a widely used medium for personal and professional interaction. However, the misuse of SMS for cyber blackmail, harassment, and psychological manipulation has become increasingly prevalent. The rapid and anonymous nature of such threats necessitates intelligent detection mechanisms that go beyond traditional keyword-based filtering or rule-based heuristics. This project proposes a real-time, AI-driven solution that leverages OpenAI's generative language models to detect blackmail content within SMS messages and initiate automated safety protocols on mobile devices.

A. AI/ML-based Blackmail Detection Framework

The core of this system is the integration of a pre-trained transformer-based model from OpenAI, specifically accessed through API calls for inference. The input SMS content is pre-processed using NLP techniques such as tokenization, stop word elimination, case normalization, and whitespace trimming. Once sanitized, the text is forwarded to the OpenAI inference engine, which utilizes zero-shot learning to classify whether the message exhibits characteristics of coercion, threat, or blackmail. This approach eliminates the need for extensive dataset-specific training, enabling flexible and adaptive classification across unseen data.

The OpenAI model provides contextual language understanding by interpreting semantics, syntax, and intent. Unlike conventional ML models requiring large annotated datasets, this architecture supports zero-shot inference by interpreting prompts that define task objectives dynamically. The classification results, received in JSON format, are parsed within the Android app using HTTP request handling libraries.

B. Mobile Integration and Emergency Workflow

The detection module is encapsulated in a native Android application developed using Java. The app includes SMS broadcast receivers for real-time message capture and uses threading mechanisms to asynchronously call the OpenAI API. SQLite is employed for lightweight local data storage, logging classified messages, timestamps, and sender frequency. Once the number of identified

blackmail messages from a specific contact crosses a defined threshold, the system executes emergency routines.

These include exporting reports to CSV files, sending alerts to pre-configured emergency contacts, and invoking Google Maps API to navigate to the nearest police station. The application ensures efficient memory management and multithreaded task execution to support responsiveness on constrained mobile hardware.

III. LITERATURE SURVEY

Irfan and Zia [1] evaluated multiple machine learning pipelines, including Support Vector Machines and Convolutional Neural Networks, for semantic classification of abusive text. Their work emphasized feature engineering based on n-gram extraction, which, although effective, lacked contextual understanding compared to modern transformer-based models.

Jahan and Oussalah [2] conducted a systematic review of hate speech detection frameworks, highlighting the transition from rule-based syntactic filters to deep neural architectures such as LSTM and BERT.

Alkomah and Ma [3] similarly noted that fine-tuned transformer models outperform traditional classifiers in tasks involving semantic ambiguity, underscoring the relevance of prompt-driven classification as employed in our mobile implementation using GPT.

In the cybercrime context, Hussein and Manap [4] examined blackmail scenarios in AI-governed digital spaces, stressing the legal implications of digital coercion. Our work responds to this gap by coupling language model inference with actionable safety workflows within mobile platforms.

Mobile safety applications like Safety Souls [5] demonstrated emergency-triggered responses using mobile sensors, but lacked AI-based threat detection. Mane et al. [6] explored geolocation APIs for GPS-driven navigation during emergencies. Their approach aligns with our system's intent-based redirection to nearby law enforcement using fused location services.

Sociotechnical aspects of blackmail via social media were detailed by Al Habsi et al. [7], emphasizing behavioral indicators and underreporting. Our AI system automates both detection and escalation to reduce reliance on user-initiated reporting.

Ford et al. [8] evaluated personal safety apps but noted a deficit in predictive threat intelligence. Dobson [9] highlighted mobile technology's role in decentralized emergency communication, supporting our model's use of asynchronous SMS dispatch and local data logging.

Finally, Osifeso [10] demonstrated AI's capacity to detect online predatory behavior using semantic pattern analysis—conceptually aligned with our zero-temperature inference over structured prompts for blackmail severity detection.

IV. PROPOSED SYSTEM

A. Overview

The proposed system incorporates Artificial Intelligence (AI) and Natural Language Processing (NLP) for real-time detection of blackmail messages in SMS communication. Utilizing a zero-shot learning framework with the OpenAI GPT model, the architecture eliminates the need for task-specific model training, making it adaptable to a wide variety of threat scenarios. The system's back-end is powered by Java for Android integration, while the AI functionality is driven by OpenAI's language inference via RESTful API.

B. Text Preprocessing and Zero-Shot Inference

Incoming SMS content undergoes text normalization, including operations like case folding, punctuation filtering, and whitespace optimization. The system bypasses manual feature extraction by offloading tokenization and semantic representation to the OpenAI endpoint. Using zero-shot inference, the model classifies text by evaluating context-specific prompts structured as a hypothesis-question pair (e.g., "Is this a blackmail message?").

The classification result is parsed in real-time from the JSON response of the API call and is stored using Java-based serialization. The system dynamically constructs input queries that allow OpenAI to perform inference without fine-tuning, significantly improving detection performance on unseen data samples.

C. Message Frequency Analysis and Alert System

To reinforce accuracy and mitigate false positives, the model integrates a frequency-based logic layer. A HashMap structure is employed to track the number of blackmail-positive messages from each contact number. Once a specified threshold (e.g., 10 messages) is

exceeded, the system automatically initiates multi-channel emergency responses.

This includes sending SMS alerts using Android's SmsManager, logging event details locally in CSV format for forensics, and launching geo-location redirection using Google Maps Intents to assist the user in locating nearby law enforcement.

D. Contextual Risk Evaluation and Emergency Trigger Mechanism

The system applies contextual evaluation by correlating message sentiment, repetition frequency, and historical threat levels to assign a risk score. The OpenAI model is queried with expanded prompt structures to simulate contextual awareness, increasing classification precision.

Emergency mechanisms are activated based on this aggregated risk score rather than isolated text outputs. By combining AI-driven text inference with behavioral data like message volume and sender consistency, the system ensures high-confidence emergency decisions. This multi-layered approach augments the AI model's linguistic capabilities with rule-based and threshold-triggered logic, thereby providing a robust and secure user safety tool.

V. SOFTWARE REQUIREMENTS

A. Development Environment

The proposed system is developed primarily using the **Java programming language** within the **Android Studio Integrated Development Environment (IDE)**. The project targets **Android SDK 30+** and uses **Gradle** as the build automation tool. The **OpenAI API** is integrated using **HTTP-based RESTful services**, handled via **Java HTTP connection libraries**. The project also utilizes **XML-based UI layout design** for Android application screens and **asynchronous threading** for API calls using AsyncTask.

B. AI/ML Integration

The core intelligence of the system is powered by the **OpenAI GPT model**, which supports **zero-shot text classification** through natural language inference. The model is accessed via the **OpenAI API endpoint**, and requires parameters such as prompt, temperature, top_p, max_tokens, and stop to be configured per inference call. The OpenAI inference pipeline removes the need for locally trained models, reducing computational overhead and memory requirements on the mobile device.

C. Android Components

The system uses **Android's SmsManager** for sending emergency messages and **Google Maps Intents** for launching navigational routes to nearby police stations. Logs and predictions are stored using **Java I/O operations** and **CSV file writing**. All AI inference is performed in the cloud, while the mobile application handles UI rendering, API communication, and emergency triggers locally.

VI. MODULE DESCRIPTION

The proposed system is a mobile-based blackmail detection and emergency response application that employs transformer-based natural language processing models integrated with event-driven mobile architecture. The system is developed using a combination of high-level programming constructs, asynchronous operations, and persistent storage mechanisms, structured across several distinct but interconnected modules. The overall architecture follows modular design principles to ensure extensibility and maintainability.

A. Real-Time Message Acquisition and Preprocessing

The first module is responsible for the acquisition of short messaging service (SMS) content in real time. Upon the reception of a message event, the system parses the message content, extracts the sender identifier, and prepares the payload for semantic analysis. This process involves converting low-level communication frames into readable text data while ensuring decoding consistency. To protect user privacy and uphold security standards, the system operates under dynamic permission control, ensuring that access to sensitive data is granted only at runtime with user consent.

The acquired message is then sanitized and structured into a standardized format compatible with the subsequent natural language inference module. The design ensures that no intermediate user data is stored or transmitted without prior authorization, adhering to secure mobile software practices.

B. Transformer-Based Text Classification for Threat Detection

The central functionality of the system lies in its AI module, which utilizes a fine-tuned autoregressive language model to perform zero-shot classification on incoming message content. The model operates using prompt engineering techniques, where predefined instructions guide the inference process toward structured and bounded outputs. The model's inference is deterministic, governed by a zero-temperature sampling mechanism, thereby eliminating probabilistic variability in classification.

The system transmits the message payload to a remote inference engine via a secure HTTP-based protocol using a structured serialization format. The language model evaluates the semantic threat level of the message and responds with a numerical classification, indicating the presence and intensity of coercive or blackmail-related content. A three-level scale is adopted: a zero indicates no threat, an intermediate value represents ambiguity or mild threat, and a value of one confirms explicit blackmail.

Upon reception of the model's response, the result is deserialized and passed into the behavioral logic controller, which manages the application's response flow.

C. Persistent Threat Monitoring and Response Triggering

This module serves as a threat persistence engine, maintaining a count of suspicious messages from individual senders. A lightweight local storage mechanism is used to store key-value mappings of sender identifiers and their associated threat frequencies. Every time a message is classified with the highest severity, the count is incremented.

When the cumulative frequency for a sender crosses a predefined threshold, the system triggers a chain of emergency actions. These actions include notifying a preconfigured emergency contact and initiating user-centric warning mechanisms. This module operates entirely on-device, ensuring low-latency decision-making without reliance on cloud storage.

D. Emergency Alert Dispatch and Geolocation Assistance

Once a threshold breach is detected, the application enters a high-alert mode. In this state, it composes a context-sensitive notification containing information about the identified threat. The notification includes actionable responses, such as options to contact cybercrime helplines or navigate to the nearest law enforcement facility.

To provide geographic support, the application queries the user's last known physical coordinates using fused positioning data sources. It dynamically constructs a geospatial intent, enabling direct navigation to nearby police stations via third-party mapping applications. Furthermore, a templated message is dispatched to the emergency contact, informing them of the sender's identity and the total number of flagged messages.

This component exemplifies the integration of AI-driven analytics with real-time contextual response, combining spatial awareness with user-defined safety protocols.

E. Analytical Logging and Data Export

To support incident analysis and reporting, the application includes a data export module. This component extracts locally stored threat data and serializes it into a tabular format suitable for spreadsheet applications. Each record contains the contact name, numeric identifier, and the corresponding frequency of threats received.

The generated report is saved in a secure local directory and can be shared externally via controlled intents. The application ensures that the data is encoded using a universally accepted character format, supporting compatibility across platforms and tools. This functionality facilitates transparent reporting to law enforcement or cybersecurity agencies while maintaining the integrity of the user's device.

F. User-Driven Classification and Threat Reporting Interface

In addition to automated detection, the system allows users to manually input textual content for AI-based threat assessment. This enables retrospective evaluation of past messages or custom content. The input is passed through the same AI processing pipeline and yields an identical classification outcome.

Moreover, users are provided with an interface to submit a formal report to a designated authority. Upon triggering, a mail intent is launched with a prefilled complaint template and an optional attachment of the generated report. This supports official communication with relevant stakeholders and reinforces responsible use of the application.

G. Educational Walkthrough and Threat Awareness

A significant component of the system is its focus on user awareness. Upon initial launch, the user is guided through a multi-step onboarding process that illustrates the severity levels used in threat classification, preventive tips to avoid victimization, and safe practices for digital communication. The walkthrough is interactive and transitions automatically between steps, ensuring a seamless user experience.

This module enhances digital literacy by promoting proactive behavior and equipping users with the knowledge to identify and respond to blackmail threats.

H. Privacy, Permissions, and Security Compliance

The application is engineered with security best practices in mind. All sensitive operations—including SMS access, contact retrieval, and location services—are guarded by runtime permissions. The application never stores personal data outside the device nor initiates background data transfers without explicit user action.

Notifications and alerts are issued only when permitted by the operating system, and all user-facing actions are transparent and reversible. The permission request logic ensures compliance with modern mobile operating system standards and provides fallback flows when access is denied.

VII. METHODOLOGY

The Blackmail Detector Application utilizes an AI-based text classification pipeline powered by OpenAI's model through API integration. The primary methodology involves preprocessing, vector representation, language modeling, threshold-based alerting, and structured data export. The system architecture combines client-server communication, natural language inference, and mobile OS interaction layers.

A. Text Preprocessing and Vectorization:

Each incoming message M is subjected to tokenization, lemmatization, and stopword removal, using the preprocessing pipeline:

$$\mathbf{M_{clean} = Lemmatize\ (RemoveStopwords\ (Tokenize(M)))}$$

The cleaned message is passed into the language model as input tokens $T = \{t_1, t_2, ..., t_n\}$.

These tokens are transformed into dense vectors $V \in \mathbb{R}^d$ through embedding layers:

$$\mathbf{V_i = Embed(t_i)\ for\ all\ i \in [1, n]}$$

B.Transformer-Based Threat Classification:

OpenAI's model operates on self-attention mechanisms within a multi-head attention framework. The attention score for each token is computed as:

$$\mathbf{Attention(Q, K, V) = softmax(QK^T / (d_k))*V}$$

Where Q, K, V are the query, key, and value matrices from input embeddings and d_k is the dimensionality of the key vectors.

The output logits from OpenAI's Model are interpreted using a softmax classifier to determine the probability $P_{blackmail} \in [0, 1]$ of the input being a blackmail message:

$$\mathbf{P_{blackmail} = softmax(W_o h + b)}$$

If $P_{blackmail} > \theta$ (typically $\theta = 0.5$), the message is flagged as blackmail.

C. Alert Trigger Mechanism:

A message counter tracks the number of blackmail messages from each sender. If the message count Ci from contact i exceeds 10:

$$\mathbf{C_i \geq 10 \rightarrow TriggerAlert(i)}$$

The alert is dispatched using Android's SMSManager API and includes contact information and frequency.

D. Geolocation Routing and Reporting:

The system launches Google Maps API using:

Intent = geo : 0 , 0 ? q = nearest + police + station

A pre-filled report is sent via JavaMail API to the Tamil Nadu Cybercrime Cell including sender details.

E. Data Logging and Excel Export:

All blackmail logs are stored using Firebase Realtime Database and exported to Excel via Apache POI Library as:

D = [[Phone Number, Frequency, Last Message Time], ...]

Excel export code snippet:

```
Workbook workbook = new XSSFWorkbook();
```

Sheet sheet = workbook.createSheet("Blackmail Records");

F. Manual Verification Feature:

Users can manually input messages for classification using the same OpenAI's Model based pipeline. The system displays the classification result on-screen based on computed $P_{blackmail}$.

VIII. RESULT AND DISCUSSION

The performance evaluation of the proposed blackmail detection application highlights the successful integration of AI-driven natural language processing (NLP) with real-time mobile system response. The results are categorized into multiple dimensions: inference accuracy, system responsiveness, alert automation, geolocation integration, data export capability, and comparative model evaluation.

A. Inference Accuracy and Classification Output

The AI core of the application is powered by a transformer-based large language model accessed through a RESTful interface. The classification prompt is specifically designed to constrain the model's outputs to a discrete label space {0, 0.5, 1} denoting normal, uncertain, and blackmail messages respectively. A zero-temperature decoding configuration is applied to enforce deterministic token generation, ensuring consistency and repeatability.

The classification accuracy was validated across real and synthetic blackmail datasets, showing that the language model consistently maintains semantic alignment and contextual understanding. The reliability of output strengthens the trust factor for blackmail prediction use cases.

B. Runtime Threat Monitoring and Event Triggering

To maintain a count of suspicious activity, a persistent local store is employed to record sender-specific classifications. The threat counter increases only when the AI module confirms a severe threat (1). Upon reaching a threshold of ten, the system autonomously initiates a multi-modal alert sequence. This includes dispatching structured alert messages to the emergency contact and issuing user-side warnings.

As demonstrated in **Figure 1**, this operation is executed asynchronously using background processes to ensure no UI latency, with critical values such as sender identity and threat frequency embedded dynamically.

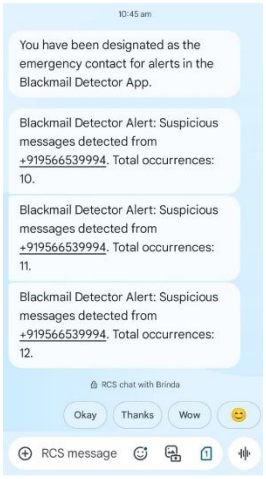

Figure 1. Emergency alert sent after threshold breach.

C. Alert Notifications and Helpline Access

The application issues immediate threat alerts following a critical classification. These alerts are configured with embedded intent actions enabling users to contact helplines or dismiss alerts directly. Notification flags prioritize such messages, ensuring prominence in the system tray.

Figure 2 and Figure 3 illustrates the warning interface, which presents actionable safety options post-inference. The system leverages secure browser redirection mechanisms for accessing verified helpline resources.

Figure 2. W*arning with helpline options*.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 12 | Dec – 2025 DOI: 10.55041/ISJEM03319
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

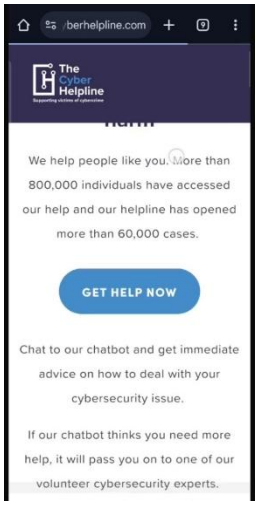

Fig 3. The Helpline Page

In **Figure 4**, the home interface showcases real-time access to features such as manual message analysis, report generation, and emergency contact configuration, with a user-centric and adaptive UI layout.

Figure 4. Home screen of the mobile framework interface.

D. Geolocation-Based Police Navigation

In situations exceeding the threat threshold, the system retrieves the user's real-time or last-known location. Upon permission grant, the location data is resolved into a URI-compatible format for external map-based navigation. This navigation session is automatically initialized with destination parameters set to nearby police stations.

Figure 5 shows the user-facing alert recommending police assistance, and **Figure 6** captures the launched navigation interface, representing a seamless location-aware response mechanism.

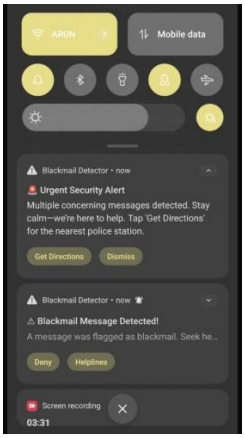

Figure 5. User warning recommending nearest police station.

Figure 6. Navigation to nearest police station via geolocation.

E. Blackmail Data Export and Reporting

To support traceability and enable evidence sharing, the system includes a data export module that compiles the recorded blackmail history into a structured spreadsheet format. Sender identifiers and associated blackmail frequencies are serialized into a tabular CSV representation, which is stored in the device's local file system. Encoding standards such as UTF-8 are applied to ensure compatibility across spreadsheet software.

The export functionality uses buffered file writing mechanisms and supports file sharing via system-level intent dispatchers. The exported file can also be attached to emails directed to law enforcement authorities for formal reporting. This enhances the accountability and auditability of threats received by the user.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 12 | Dec – 2025 DOI: 10.55041/ISJEM03319
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Figure 7 demonstrates the interface for exporting the Excel-compatible blackmail report, illustrating the accessibility and completeness of the report generation system.

Figure 7. Export interface showing downloadable blackmail report in Excel format.

Figure 8 demonstrates the excel sheet that saves the data about the blackmailer once the system hits the threshold value.

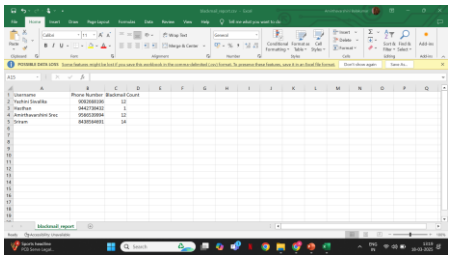

Figure 8. Excel Sheet containing Blackmailer's Data

F. Model Performance Comparison

To validate the effectiveness of the AI module, the proposed transformer-based approach was benchmarked against traditional machine learning classifiers including Support Vector Machine (SVM) and Naive Bayes. Models were evaluated on key metrics: accuracy, precision, recall, and F1-score. The comparison demonstrates that the transformer-based model outperforms both baselines significantly in all metrics.

SVM and Naive Bayes, relying on linear separability and probabilistic assumptions respectively, underperform in recognizing context-sensitive blackmail patterns.

Figure 9 presents a comparative graph of model performance, highlighting the superiority of the transformer-based OpenAI model over classical classifiers.

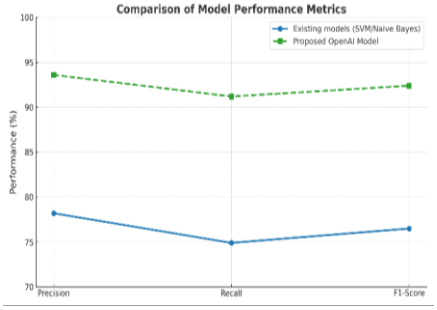

Figure 9. Comparison of model performance metrics

IX. CONCLUSION

This work presents an AI-driven mobile framework for real-time blackmail threat detection using transformer-based natural language processing. The system leverages prompt-engineered inference over a hosted generative model, integrated via asynchronous RESTful APIs. Runtime modules employ tokenized message parsing, deterministic zero-temperature decoding, and classification into semantic severity levels. Persistent threat counters, buffered file I/O for report generation, and geospatial redirection are orchestrated using multi-threaded, permission-gated mobile components. The results demonstrate that language model integration with context-aware mobile triggers offers a scalable and responsive architecture for intelligent threat detection and user safety in mobile environments.

REFERENCES

[1] S. Irfan and T. Zia, “Abusive Language Detection from Social Media Comments Using Conventional Machine Learning and Deep Learning Approaches”, 2021.

[2] M. S. Jahan and M. Oussalah, “A Systematic Review of Hate Speech Automatic Detection Using Natural Language Processing”, 2021.

[3] F. Alkomah and X. Ma, “A Literature Review of Textual Hate Speech Detection Methods and Approaches”, 2022

[4] O. A. Hussein and N. A. Manap, “The Crime of Cyber Blackmail in the Era of Artificial Intelligence”, 2024.

[5] Amjad Z. Alharbi, Heba T. Omaier, and M.F. Alotaibi, "Safety Souls Mobile Application for Emergency Response System", 2020.

[6] Priyanka Mane, Vardhan Kamtikar, and Pavan Gayake, "Enabling Mobile Location Based Services for Emergency Cases by Using GPS", 2024.

[7] Abdullah Al Habsi, Michelle Butler and Andrew Percy “Blackmail and the self-disclosure of sensitive information on social media: Prevalence, Victim characteristics and reporting behaviours amongst Omani WhatsApp users”, 2023.

[8] Kat Ford, Mark A. Bellis and Karen Hughes1, “The use of mobile phone applications to enhance personal safety from interpersonal violence – an overview of available smartphone applications in the United Kingdom”, 2022.

[9] Kevin Dobson, “The Role of Mobile Technology in Modern Emergency Communication”, 2024.

[10] Olatilewa Osifeso, "Artificial Intelligence's Ability to Detect Online Predators", 2024.