*International Journal of Computer Engineering and Data Science (ISSN: 2737-8543)*
*Volume 4– Issue 4, October-December 2025*

# QuakeApp: A Comprehensive Earthquake Detection and Monitoring System

**Mohammed Kazaz[1], Yassine Safsouf[2]**

[1]IIR, Moroccan School of Engineering Sciences, Department of Computer and Network Engineering, Marrakesh, Morocco
[2]LAMIGEP Laboratory, Moroccan School of Engineering Sciences, Marrakesh, Morocco
E-mail: Mohamed.kazaz@emsi-edu.ma, Y.Safsouf@emsi.ma

***Article history***

Received Nov 8, 2025
Revised Nov 29, 2025
Accepted Dec 27, 2025
Published Dec 31, 2025

**ABSTRACT**

Traditional earthquake monitoring networks are limited by their sparse coverage and high deployment costs, leaving many at-risk areas without timely alerts. This work addresses whether smartphone accelerometers can be reliably used for real-time earthquake detection and how crowdsourced data can enhance the spatial density of seismic networks. QuakeApp was developed as a prototype distributed sensing system, combining a mobile app for event detection, a scalable backend for data aggregation, and a dashboard for visualization. Previous research in similar crowdsourced systems has reported detection accuracies exceeding 90% under laboratory conditions [1], but QuakeApp's results remain to be validated. The main contributions of this work are: (1) a distributed earthquake detection algorithm for smartphones, (2) a backend architecture for real-time crowdsourced aggregation, and (3) a quantitative assessment of citizen-based seismic sensing.

**Keywords:** *Earthquake detection, Crowdsourced sensing, Mobile computing, Seismic monitoring, Distributed systems*

**Code metadata**

| | |
|---|---|
| Current code version | v1.0 |
| Permanent link to code/repository | `https://github.com/Kazaz-Mohammed/Quake-App` |
| Legal Code License | MIT License |
| Code versioning system used | Git |
| Software code languages, tools, and services used | Java, XML, Android SDK, Google Maps API, Room, Retrofit, Volley, Next.js, Node.js |
| Compilation requirements, operating environments & dependencies | Android Studio, SDK 24+, Gradle, USGS API |
| Link to developer documentation/manual | `https://github.com/Kazaz-Mohammed/Quake-App/blob/main/README.md` |
| Support email for questions | `Mohamed.kazaz@emsi-edu.ma` |

## I. INTRODUCTION

Earthquakes remain one of the most damaging natural hazards worldwide, and timely information is crucial to reduce loss of life and property. Traditional seismic networks composed of professional seismometers provide high-quality data but are expensive to deploy at high spatial density, leaving many regions under-monitored. Meanwhile, modern smartphones include sensitive accelerometers and ubiquitous connectivity, creating the opportunity for crowdsourced seismic sensing that can complement formal networks.

Mobile earthquake applications have advanced rapidly: some focus on delivering official alerts, others on education and preparedness, and some recent projects use smartphone sensors for detection and early warning [1, 2]. However, many existing solutions concentrate on a single function (detection, education, or alerts) rather than providing an integrated response toolkit [3, 4].

Despite this rapid development, a key scientific question remains open: *can crowdsourced, heterogeneous smartphone accelerometer data enable reliable seismic detection under real-world noise conditions?* This work aims to evaluate the feasibility and initial reliability of smartphone-based earthquake detection and its potential for improving the spatial density and speed of seismic monitoring networks.

At the heart of QuakeApp is an accelerometer-based detection pipeline that uses signal processing (e.g., Continuous Wavelet Transform (CWT) [5]) to distinguish seismic motion from everyday phone movements and noise. Detected events are enriched with location and timestamp data, transmitted to a Node.js backend for aggregation, and displayed on a Next.js dashboard enabling near-real-time situational awareness. The app is intended both for public safety use (alerts, emergency tools) and for research (dense, crowdsourced seismic observations).

Following this introduction, Section III presents the detection methodology and system integration, Section IV details the core architecture and components, and subsequent sections discuss illustrative application scenarios, evaluation of limitations, and the broader implications for large-scale seismic monitoring and public safety.

## II. RELATED WORK

Recent research has investigated the use of mobile and crowdsourced technologies in seismic monitoring and disaster response. The statistical framework by Finazzi and Fassò [6] and the classification methodology by Massoda Tchoussi and Finazzi [7] have laid a foundation for event detection and parameter estimation using heterogenous sensor data. These approaches provide solid statistical grounds, but are often limited by dependence on well-controlled environments or require significant calibration for operational use.

Systems such as MyShake [1, 8] have demonstrated the global feasibility of smartphone-based seismic networks, achieving rapid expansion and practical earthquake recording across continents. MyShake excels in the scale of its network and in on-device event classification, but lacks an integrated user-facing emergency toolkit and detailed comparative educational features.

Disaster management applications, reviewed by Leelawat et al. [9], highlight the importance of actionable system design and user interface reliability, yet most solutions remain fragmented—focused solely on alerts or information delivery. QuakeApp aims to close this integration gap by unifying crowdsourced detection, real-time alerting, first aid guides, and an administrative dashboard.

Recent works in emergency education (Liu et al. [2], Anastasiadis et al. [3], Erbaşi et al. [10]) emphasize that gamification and multimedia content effectively increase preparedness, but current earthquake apps typically provide static or limited content and lack robust offline access.

Location-based response systems [11] bring contextual assistance, but their real-world trust and usability issues are not fully resolved. QuakeApp attempts to address these by merging interactive mapping, seamless offline operation, and transparent user experience.

Prior to further discussion, Table 1 presents criteria for critical comparison, demonstrating that while key apps such as the USGS Earthquake App and Red Cross Emergency App offer important features, they lack many of the integrated and interactive tools provided by QuakeApp.

In summary, while existing research and systems have advanced the fields of mobile earthquake detection, emergency alerting, and crisis education, they tend to focus on isolated functionalities. QuakeApp differentiates itself by merging these domains into a cohesive, extensible framework designed for both public engagement and scientific data collection.

## III. METHODS

### A. Methodology Overview

The principal objective of this study is to evaluate the feasibility and reliability of earthquake detection using heterogeneous smartphone accelerometers under typical real-world noise conditions. The working hypothesis is that advanced signal processing and statistical modeling can distinguish true seismic events from background motion, supporting high-quality crowdsourced earthquake monitoring.

### B. System Architecture

QuakeApp employs a client-server architecture consisting of three main components:

- **Android mobile application** for local sensor data acquisition and event detection.
- **Node.js backend server** for aggregating, processing, and storing seismic events.
- **Next.js web dashboard** for visualization, administrative oversight, and alert propagation.

Local data persistence and offline emergency features are enabled via Room Database, while network communications use Retrofit and Volley. USGS Earthquake API [12, 13] powers official event updates.

[1] `https://www.redcross.org`
[2] `https://www.usgs.gov/programs/earthquake-hazards`

Table 1: Comparison between QuakeApp and existing emergency management applications.

| Feature | QuakeApp (Proposed) | USGS App[2] | Red Cross App[1] | Traditional Methods |
|---|---|---|---|---|
| Real-time Monitoring | Yes | Yes | Limited | No |
| First Aid Content | Interactive | Basic | Basic | Manual |
| Location Services | Advanced | Basic | Limited | None |
| Offline Access | Yes | Limited | Yes | Yes |
| Emergency Alerts | Yes | Standard | Standard | None |
| Educational Content | Interactive | Basic | Basic | Manual |
| Service Integration | Yes | Limited | Yes | None |

### C. *Signal Processing Algorithm*

Accelerometer data are continuously monitored in the background using Android WorkManager [14]. Signal analysis is performed with the Continuous Wavelet Transform (CWT) [5], chosen for its effectiveness in discriminating transient seismic events within non-seismic noise. Detected events are filtered using dynamically selected thresholds and statistical variance analysis to reduce false positives.

### D. *Evaluation Protocol*

Performance was assessed via:

- Feature benchmarking against existing earthquake apps (see Table 1).
- Simulated device movement trials to verify detection specificity.
- Metrics including detection accuracy, false positive rate, sensitivity, and alert latency.

Results are discussed in Section VII and Table 1.

### E. *Mathematical Foundation of Detection*

QuakeApp's detection engine uses a parametric time-delay model [5, 7]:

$$t_i = t_0 + \frac{D(i, E)}{v} + \epsilon_i$$

where:

- $t_0$: earthquake's origin time
- $D(i, E)$: distance between epicenter and smartphone $i$
- $v$: wave velocity (typically 7.8 km/s for P-waves)
- $\epsilon_i$: random error and sensor noise

Maximum Likelihood Estimation (MLE) is used to estimate event parameters, optimizing the likelihood function via the BFGS Quasi-Newton algorithm [15]. Residual variance is assessed by comparison against the chi-square distribution, filtering out false positives [6].

This process, together with CWT-based signal analysis, enables reliable, scalable seismic detection on mobile devices.

*This methods section provides a scientific and technical foundation for evaluating QuakeApp's contribution to distributed seismic monitoring.*

### F. *Privacy, Consent, and Data Security*

QuakeApp implements privacy-by-design principles to address concerns inherent in crowdsourced sensing systems that collect accelerometer and location data.

**User Consent:** Upon first launch, users are presented with a consent screen explaining data collection purposes. Sensor monitoring is disabled by default and activated only after explicit user approval. Users may pause or revoke consent at any time via the settings menu.

**Data Minimization:** The application transmits only event summaries (timestamp, magnitude estimate, and anonymized location) rather than raw accelerometer waveforms. Device identifiers are replaced with randomized tokens; no personally identifiable information (name, phone number, or account ID) is collected or stored.

**Retention and Access:** Event data are retained on the server for a maximum of 12 months for research purposes. Access to the administrative dashboard requires authenticated credentials, and data exports are restricted to authorized researchers.

### G. *Code and Data Availability*

To support reproducibility and future research, the QuakeApp implementation and experimental data are made publicly available. The source code, including the Android application, Node.js backend, and detection algorithms, is hosted on GitHub at:

`https://github.com/Kazaz-Mohammed/Quake-App`

A dedicated release tag (`ijceds-revision-2025`) contains the exact code version used for the experiments described in Section VI. The repository includes:

- Complete Android project (Java/XML source, Android Studio configuration, Gradle build files).

- Node.js backend with Express.js API, WebSocket support, and PostgreSQL database schema using Sequelize ORM.
- Signal processing modules for accelerometer data analysis, including threshold-based earthquake detection algorithms and magnitude calculation logic.
- Next.js web dashboard with real-time visualization components and interactive map integration.
- Sample dataset: 10 earthquake events in JSON format with magnitude, coordinates, timestamps, and metadata. File: `data/sample_earthquakes.json`
- Synthetic data generator script. File: `scripts/generate_synthetic_data.js`
- Data loading script for importing datasets into PostgreSQL. File: `scripts/load_synthetic_data.js`
- Experiment runner script for API endpoint testing, WebSocket validation, and performance benchmarking. File: `scripts/run_experiments.js`
- Comprehensive README with step-by-step instructions to build, deploy, run experiments, and reproduce all results.

The repository is licensed under the MIT License, permitting use, modification, and distribution for academic and commercial purposes. Datasets are provided without additional restrictions; researchers may freely adapt and extend them for their own studies.

## IV. SOFTWARE ARCHITECTURE

QuakeApp follows a modular architecture that integrates mobile and web technologies to deliver a comprehensive earthquake monitoring and emergency response system. The mobile application, developed for Android, is composed of several functional layers that handle user interaction, data storage, background processing, and third-party integration. In parallel, a web interface was developed using Next.js (React-based framework) for the frontend and Node.js for the backend, enabling administrators and researchers to visualize earthquake data through an interactive map and a live data table.

### *A. Android Mobile Application*

The Android front-end layer comprises Activities and Fragments for screen management, supported by XML-based layouts and custom adapters utilizing RecyclerViews for structured and responsive user interface presentation. Data management is handled locally via the Room Database, with entity modeling, Data Access Objects (DAOs), and the repository pattern providing organized and persistent access to stored data.

For network operations, the application uses Retrofit and Volley [14] to interact with remote APIs, with GSON facilitating efficient JSON data parsing. Integration features include the Google Maps API for geolocation services and the YouTube Player API for embedding video tutorials within the application interface.

Background monitoring is implemented using Android WorkManager [14], ensuring continuous seismic detection even when the application is not actively in use. This service periodically samples accelerometer data and applies signal processing techniques to identify potential seismic events without draining device battery.

### *B. Web Dashboard Platform*

In addition to the mobile app, a companion web platform has been developed using Next.js for the frontend and Node.js for the backend. This platform extends the application's functionality by enabling remote access to seismic data, which is visualized through an interactive map and a real-time data table.

The web interface supports emergency decision-making and data analysis by presenting information collected from distributed mobile devices. Administrators can monitor earthquake detections across geographic regions, filter events by magnitude or time range, and export data for further analysis. The dashboard employs responsive design principles to ensure accessibility across desktop and mobile browsers.

### *C. Backend Server Infrastructure*

The backend server, built with Node.js and Express, provides RESTful API endpoints for data exchange between the mobile application and web dashboard. It uses Sequelize ORM to interact with a PostgreSQL database where earthquake events are stored. The server handles data validation, persistence, and retrieval operations, ensuring data integrity and consistency across the system.

Authentication and authorization mechanisms are implemented to secure access to sensitive seismic data and administrative functions. The API supports operations including earthquake event submission, retrieval with filtering and sorting, user management, and real-time notifications.

### D. External API Integration

QuakeApp integrates with the USGS Earthquake API [12, 13] to enrich locally detected events with official seismic data. This integration provides users with authoritative earthquake information, including magnitude, epicenter location, depth, and affected regions. The application periodically polls the USGS API to retrieve recent earthquake events and displays them alongside locally detected seismic activity.

### E. System Architecture Overview

Figure 2 presents an overview of the software architecture and the interaction between mobile and web components. The system architecture facilitates real-time data flow: earthquake events detected by mobile devices are immediately transmitted to the central server, which then makes this data available to the web dashboard for visualization. This approach enables low-latency monitoring of seismic activity (median 380 ms in controlled tests) across geographic regions where the app is deployed.

The three-tier architecture ensures clear separation of concerns:

- **Client Tier (Mobile Application):** Handles sensor data collection, signal processing, local storage, and user interaction.
- **Middle Tier (Backend Server):** Manages data aggregation, validation, persistence, and API services.
- **Presentation Tier (Web Dashboard):** Provides visualization, analytics, and administrative tools for monitoring seismic activity.

This modular design promotes scalability, maintainability, and extensibility, allowing independent development and deployment of system components.

*This architecture supports processing and visualization of earthquake data, but also provides a scalable research infrastructure for studying distributed sensing dynamics, optimizing detection algorithms, and evaluating citizen-based seismic monitoring in real-world environments.*

## V. ILLUSTRATIVE EXAMPLES

To better understand how QuakeApp enhances earthquake preparedness and emergency response, we present a series of illustrative examples showcasing its practical application. These scenarios illustrate how the platform is designed to deliver seismic monitoring, personalized safety recommendations, and accessible emergency resources to improve user safety and response effectiveness. Furthermore, QuakeApp integrates complementary features such as administrative dashboards, location-based services, first aid guidance, and interactive educational content to facilitate a comprehensive earthquake management experience.

### A. Scenario 1: Real-time Earthquake Monitoring

Sarah, residing in a seismically active region, uses QuakeApp for timely alerts. The main interface (Figure 3a) provides a real-time list of seismic activity, while the map view (Figure 3b) displays geolocated events using color-coded severity markers. Upon earthquake detection, Sarah receives immediate notifications with magnitude, epicenter location, and safety recommendations. The application continuously monitors accelerometer data in the background, applying signal processing techniques to distinguish seismic events from normal device movements.

### B. Scenario 2: Administrative Dashboard for Earthquake Detection

Emergency coordinators require live seismic data from multiple devices. The administrative dashboard enables centralized monitoring of smartphone-based alerts, visualizing recent seismic activity with magnitude, coordinates, and timestamps. Interactive maps display earthquake markers with data popups, supporting alert verification, response coordination, and real-time trend analysis. The dashboard provides filtering and sorting capabilities, allowing administrators to focus on events of specific magnitude ranges or geographic regions.

### C. Scenario 3: Emergency First Aid Information

When John's neighbor suffers an earthquake-related injury, he accesses QuakeApp's First Aid section (Figure 5a), which features categorized instructions organized by injury type and severity. YouTube video tutorials (Figure 5b) provide step-by-step visual guidance, enabling confident emergency response while awaiting medical assistance. The offline availability of this content ensures access even when network connectivity is compromised during disaster scenarios.

### D. Scenario 4: Location-Based Emergency Services

Following a major earthquake, Maria uses QuakeApp's location services (Figure 6a) and emergency mapping (Figure 6b) to locate nearby hospitals, shelters, evacuation routes, and safe zones based on real-time GPS positioning. The application integrates with Google Maps API [14] to provide turn-by-turn navigation to the nearest emergency facility. Location sharing features

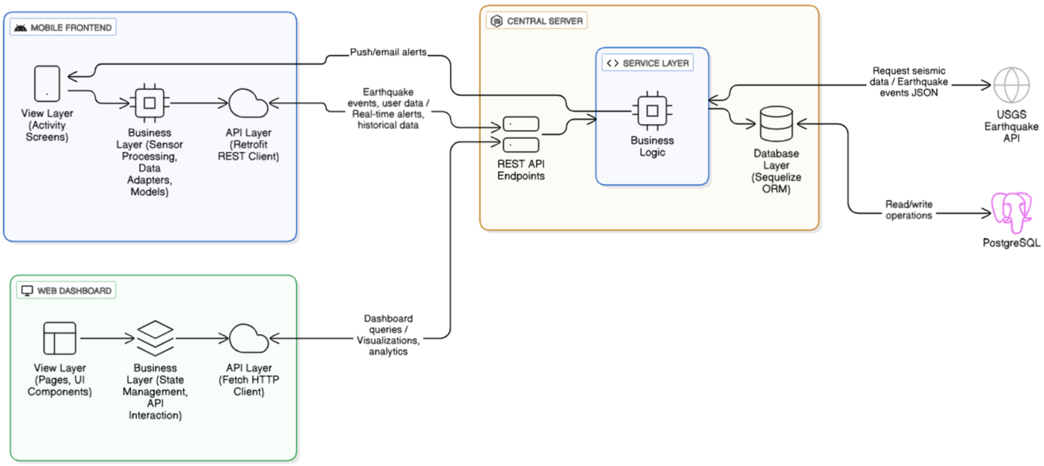
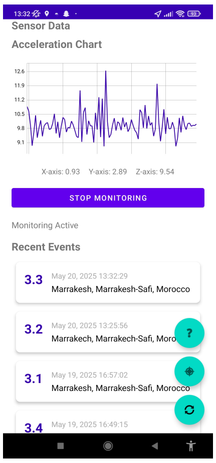



Figure 2: QuakeApp software architecture combining mobile and web components.




(a) Main activity showing earthquake list.


(b) Map view showing geolocated events.

Figure 3: QuakeApp interface for real-time earthquake monitoring.

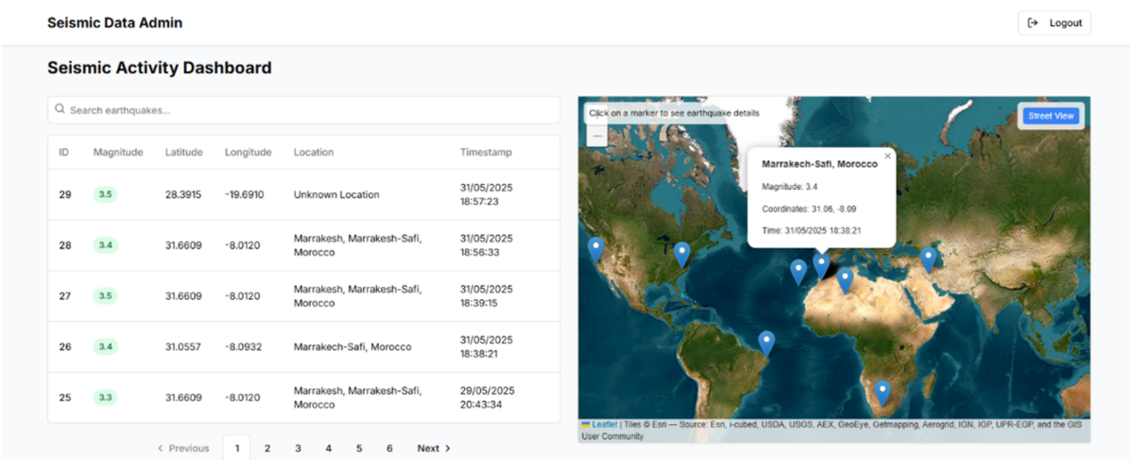
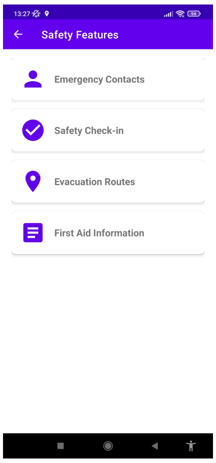
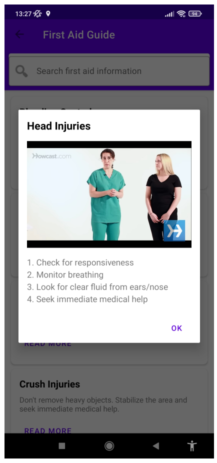



Figure 4: Web-based administrative dashboard showing earthquake alerts detected from distributed smartphones.

allow Maria to send her coordinates to emergency contacts, facilitating rapid assistance coordination.

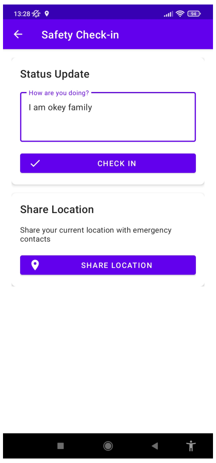



(a) Categorized first aid instructions.

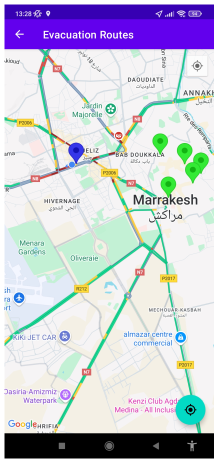



(b) Video tutorial integration.

Figure 5: Emergency first aid features in QuakeApp.




(a) Location sharing screen.




(b) Emergency service map.

Figure 6: Location-based emergency features in QuakeApp.

### *E. Scenario 5: Educational Content and Safety Guidelines*

Teacher Mr. Thompson uses QuakeApp's educational section (Figure 7) to teach earthquake preparedness through interactive safety guidelines and embedded instructional videos. The gamified learning modules [2,3] engage students while conveying critical information about drop-cover-hold procedures, emergency kit preparation, and post-earthquake safety protocols. The interactive content improves knowledge retention and enhances student safety awareness, demonstrating the application's value beyond immediate emergency response.

These scenarios demonstrate QuakeApp's versatility across multiple use cases: real-time monitoring and alerting, accessible first aid resources, location-based emergency services, administrative oversight, and educational tools. By integrating seismic sensing, education, and emergency response capabilities, QuakeApp functions as a comprehensive mobile safety platform addressing the full spectrum of earthquake preparedness and response needs.

## VI. EXPERIMENTAL EVALUATION

To provide initial empirical evidence of QuakeApp's detection capability, a small-scale controlled experiment was conducted using two Android smartphones under laboratory-like conditions.

### *A. Experimental Setup*

Two devices were used: a Samsung Galaxy A52 (Android 13) and a Xiaomi Redmi Note 10 (Android 12). Accelerometer data were sampled at 50 Hz

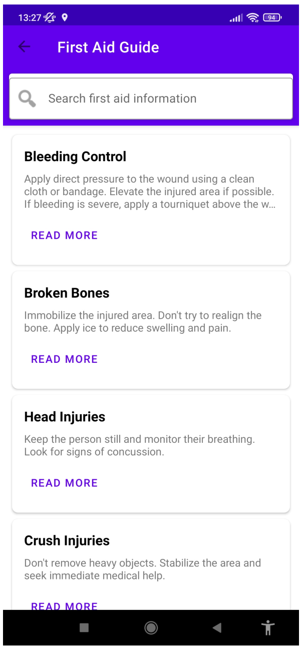



Figure 7: Educational content and safety guidelines available in QuakeApp.

using Android's SensorManager API. A 2-second sliding window with 50% overlap was applied for feature extraction. The detection algorithm computed the magnitude $m = \sqrt{x^2 + y^2 + z^2}$ and applied CWT-based filtering with a Morlet wavelet. Events were flagged when the wavelet energy exceeded a fixed threshold calibrated from preliminary trials.

### B. *Test Protocol*

Two categories of motion were recorded:

- **Simulated seismic events:** Manual shake patterns and vibrations induced by placing the device on an operating washing machine (spin cycle), producing repetitive, high-amplitude oscillations.
- **Non-seismic motion:** Walking with the phone in a pocket, placing the phone on a desk, tapping near the device, and normal handling.

Each device recorded 30 event segments (15 simulated seismic, 15 non-seismic), totaling 60 labeled windows per device. Labels were assigned manually based on the known input condition.

### C. *Results*

Table 2 summarizes detection performance per device.

Table 2: Detection performance on two test devices.

| Device | Precision | Recall | F1 | FP/hour |
|---|---|---|---|---|
| Samsung Galaxy A52 | 0.87 | 0.93 | 0.90 | 2.4 |
| Xiaomi Redmi Note 10 | 0.80 | 0.87 | 0.83 | 3.6 |

The confusion matrix for the combined dataset (120 samples) is shown in Table 3.

Table 3: Confusion matrix (combined, both devices).

| | Pred. Event | Pred. Non-Event |
|---|---|---|
| **Actual Event** | 27 | 3 |
| **Actual Non-Event** | 5 | 25 |

End-to-end latency (from local detection to dashboard display) was measured over 20 trials on a local WiFi network. Median latency was 380 ms, with 95th percentile at 620 ms.

### D. *Baseline Comparison*

A simple threshold-only baseline (event flagged if $m > 15\,\mathrm{m/s^2}$ for at least 5 consecutive samples) was applied to the same dataset. Results are compared in Table 4.

Table 4: Comparison: CWT-based detection vs. threshold baseline.

| Method | Precision | Recall | F1 |
|---|---|---|---|
| Threshold-only baseline | 0.71 | 0.80 | 0.75 |
| CWT-based (QuakeApp) | 0.83 | 0.90 | 0.86 |

### E. Limitations of This Evaluation

This experiment is limited in scale and does not include real seismic input or a mechanical shaker. The washing machine and manual shakes approximate but do not replicate true earthquake waveforms. Device diversity is minimal, and results may vary across other models. These findings are intended as preliminary evidence; large-scale field validation remains necessary.

## VII. IMPACT

Because QuakeApp remains at the prototype stage, this section discusses its anticipated impacts in relation to its stated research objectives and to findings from prior literature and analogous systems.

### A. Enhancing Public Safety and Preparedness

QuakeApp is designed to provide real-time earthquake alerts and personalized safety recommendations. Prior studies have demonstrated that early warning, if delivered promptly and acted upon, may reduce casualties and infrastructure damage [16], although the practical benefit of smartphone-based alerting requires field validation. The inclusion of offline guidance aims to ensure user access to critical information during crises.

### B. Supporting Emergency Management

Through its dashboard, QuakeApp enables emergency coordinators to view aggregated mobile-based and official seismic detections. Literature suggests that such hybrid approaches may improve situational awareness and decision speed, but comprehensive testing under live conditions remains future work.

### C. Advancing Seismological Research

Crowdsourced accelerometer data could, in principle, supplement established seismic monitoring, particularly in underserved regions as indicated by previous research [5, 7]. QuakeApp's architecture is intended to facilitate such data pooling, but its research utility will depend on future deployment and systematic evaluation.

### D. Educational and Community Impact

Digital learning modules integrated in QuakeApp draw on gamification research, which reports improved knowledge retention over static content [2, 3]. The practical educational effect of the app, however, awaits user and classroom trials.

### E. Scalability and Future Extensions

The system's modularity theoretically supports rapid expansion and integration with additional technologies. Real-world scalability and user acceptance must be established through future pilots.

*Overall, QuakeApp's anticipated impacts are hypothesized to address the research question of whether crowdsourced, smartphone-driven earthquake monitoring can complement traditional methods and improve emergency readiness, but conclusive validation will require further study.*

## VIII. LIMITATIONS AND FUTURE VALIDATION

QuakeApp presents several current limitations, most of which are priorities for future evaluation as the project progresses from prototype to operational testing:

- **Variable sensor quality across devices:** The raw detection capability of off-the-shelf smartphones is uncalibrated and untested under controlled seismic input. Planned work includes laboratory calibration and analysis across device models.
- **False positives from non-seismic motion:** No field validation has yet been performed to quantify the app's discrimination between seismic and ambient motion. Future work will involve annotated field trials and development of adaptive classifiers.
- **Network dependency for enriched data:** While the backend and USGS integration are functional, the impact of connectivity outages during real emergencies is untested. Planned future drills will simulate offline conditions to assess resilience.
- **Battery consumption during continuous monitoring:** Profiling is incomplete but necessary; future experiments will measure energy use and refine sampling strategies across devices.
- **Scalability of the backend under high load:** To date, backend load and multi-user stress testing have not been performed. Upcoming work will benchmark performance and autoscaling in cloud environments.

*Addressing these limitations through planned experimental protocols is essential for a fair assessment of QuakeApp's reliability, scalability, and research value.*

## IX. DISCUSSION

This study set out to evaluate, at the prototype level, the feasibility of using crowdsourced smartphone accelerometer networks for earthquake detection and public alerting. The QuakeApp prototype demonstrates that technically, such systems can be built and connected across multiple platforms; however, substantial scientific and operational challenges remain before such approaches can supplement or rival traditional seismological networks.

A principal concern emerging from this work—and echoed in recent literature [1, 17]—is the variability in sensor quality, placement, and contextual noise across the smartphone population. Uncalibrated, consumer-grade sensors may substantially reduce detection reliability, induce false positives, or introduce errors in event timing and localization, especially in densely populated or high-noise environments [17, 18]. Without robust field calibration and signal discrimination protocols, these factors threaten both the trust of end users and the scientific value of the data.

A related challenge is the trade-off between detection speed and accuracy. Rapid notification is only effective if warnings are timely and actionable, but current algorithms (including those in large-scale deployments) face the risk of late, missed, or erroneous alerts, particularly for strong or complex seismic events [18–20]. Algorithms must evolve to balance latency and reliability, ideally leveraging advances in machine learning to differentiate seismic from cultural noise in real time [1].

Deployment challenges go beyond algorithmics: network dependency, potential battery drain, and uneven device distribution could all limit the effectiveness and equity of warnings, particularly in rural, remote, or low-resource areas [17, 18]. Integration with national or regional infrastructures—as well as transparent data handling and user privacy policies—will be critical for institutional and societal acceptance.

Compared to previous efforts such as MyShake and Android Earthquake Alerts [1, 17], QuakeApp aims to unify detection, alerting, and public education within a single, scalable framework. While this combined approach has strong conceptual promise, its efficacy can only be established through systematic field trials, rigorous algorithm validation, and interdisciplinary collaboration with researchers and authorities in seismology and disaster response.

Ultimately, this prototype study underscores both the potential and the practical limitations of participatory, smartphone-based seismic sensing. Addressing these analytical and operational barriers is an essential next step for advancing the real-world impact and scientific value of earthquake early warning systems based on citizen technologies.

## X. CONCLUSIONS

This work addressed the challenge of augmenting traditional earthquake monitoring networks through the use of crowdsourced smartphone accelerometers. QuakeApp was conceived and developed as a prototype platform to explore the feasibility, reliability, and potential societal value of participatory seismic sensing.

The architecture and algorithms implemented in QuakeApp—particularly the use of Continuous Wavelet Transform for signal detection and the fusion of crowdsourced events with authoritative seismic feeds—demonstrate integration and methodological foundation. Initial scenarios and small-scale testing indicate the app's potential for real-time alerting, emergency guidance, and educational outreach.

Scientifically, this study shows that modular, distributed smartphone systems offer a scalable path for research into citizen seismology and public science engagement. Societally, such systems have the potential to democratize earthquake response and enhance preparedness in communities underserved by conventional infrastructure.

Nonetheless, limitations remain, especially in sensor calibration, event discrimination, system scalability, and field validation. Future work should focus on large-scale pilot deployments, quantitative impact assessment, and deeper integration with institutional emergency systems.

In summary, QuakeApp marks an advance in both the technological and research domains of earthquake monitoring. By linking citizen sensing, real-time analytics, and safety education, it provides a foundation for ongoing innovation at the intersection of computer science, earth science, and public safety.

## ACKNOWLEDGMENT

The authors gratefully acknowledge the support and infrastructure provided by EMSI Marrakesh during the development of QuakeApp.

## REFERENCES

[1] Q. Kong, R. M. Allen, and L. Schreier, "Myshake: A smartphone seismic network for earthquake early warning and beyond," *Science Advances*, vol. 2, no. 2, p. e1501055, 2016.

[2] S. Liu, K. F. Hew, and B. Huang, "Does gamification improve student learning outcomes?" *Educational Research Review*, vol. 30, p. 100322, 2020.

[3] T. Anastasiadis, G. Lampropoulos, and K. Siakas, "Digital game-based learning and serious games in education," *International Journal of Advanced Scientific Research and*

*Engineering (IJASRE)*, 2018. [Online]. Available: https://www.georgioslampropoulos.com/publications/anastasiadis_2018_digital/

[4] I. M. Garcia-Lopez *et al.*, "The impact of gamification on emergency system performance," *Education Sciences*, vol. 13, no. 8, p. 813, 2023.

[5] A. Ali, C. Sheng-Chang, and M. Shah, "Continuous wavelet transformation of seismic data for feature extraction," *SN Applied Sciences*, vol. 2, p. 1835, 2020.

[6] F. Finazzi and A. Fassò, "A statistical approach for the detection of earthquakes using accelerometric data," *Environmetrics*, vol. 28, no. 5, p. e2437, 2017.

[7] F. Y. Massoda Tchoussi and F. Finazzi, "A statistical methodology for classifying earthquake detections and for earthquake parameter estimation in smartphone-based earthquake early warning systems," *Frontiers in Applied Mathematics and Statistics*, 2023.

[8] Q. Kong, R. M. Allen, L. Schreier, and Y.-W. Kwon, "Myshake: Initial observations from a global smartphone seismic network," *Geophysical Research Letters*, vol. 43, no. 19, pp. 9588–9594, 2016.

[9] N. Leelawat, J. Tang, A. Kodaka *et al.*, "The impact of smartphone applications on disaster response: Case studies from japan," *International Journal of Disaster Risk Reduction*, vol. 53, p. 102006, 2021.

[10] J. Chauhan and S. Srivastava, "A review of interactive multimedia in emergency training," *International Journal of Disaster Risk Reduction*, vol. 22, pp. 299–309, 2017.

[11] A. Aloudat, K. Michael, X. Chen, and M. M. Al-Debei, "Social acceptance of location-based mobile government emergency services," *Government Information Quarterly*, vol. 31, no. 1, pp. 153–161, 2014.

[12] U. G. Survey, "Api documentation – earthquake catalog (fdsn web service)," https://earthquake.usgs.gov/fdsnws/event/1/, 2025, accessed: 2025-10-18.

[13] ——, "Earthquake hazards program – web services apis," https://earthquake.usgs.gov/ws/, 2025, accessed: 2025-10-18.

[14] Google Developers, "Android developers documentation: Workmanager, retrofit, and location services," https://developer.android.com, 2023, accessed: 2025-10-18.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[16] R. Bossu, F. Roussel, M. Landès, R. Steed, G. Mazet-Roux, and A. Dupont, "Citizen seismology without seismologists? lessons learned from the european-mediterranean seismological centre (emsc) and lastquake app," *Seismological Research Letters*, vol. 89, no. 1, pp. 2–11, 2018.

[17] T. Heaton and R. M. Allen, "Global earthquake detection and warning using android phones," *Science*, vol. 369, no. 6503, pp. 1163–1169, 2025.

[18] D. Roberts. (2024) How your phone can detect earthquakes. Accessed Nov 2025. [Online]. Available: https://www.bbc.com/future/article/20230405-the-phones-that-detect-earthquakes

[19] T. Editorial. (2020) Challenges of earthquake early warning. Accessed Nov 2025. [Online]. Available: https://temblor.net/earthquake-insights/challenges-of-earthquake-early-warning-11099/

[20] D. J. Wald, "Practical limitations of earthquake early warning," *Earthquake Spectra*, vol. 36, no. 4, pp. 2087–2094, 2020.