ResQMesh: A Hybrid Infrastructure-Independent Disaster Communication and SOS Verification Framework

Akshath Narvekar
Computer Engineering Department
Fr. Conceicao Rodrigues Institute of Technology
University of Mumbai, Vashi, India
akshath.narvekar02@gmail.com

Subhransu Patro
Computer Engineering Department
Fr. Conceicao Rodrigues Institute of Technology
University of Mumbai, Vashi, India
subhransupatro061@gmail.com

Sandra Kappani
Computer Engineering Department
Fr. Conceicao Rodrigues Institute of Technology
University of Mumbai, Vashi, India
sandra.subeesh211004@gmail.com

Amit Singh
Computer Engineering Department
Fr. Conceicao Rodrigues Institute of Technology
University of Mumbai, Vashi, India
amitrajputdeonia00@gmail.com

Bhakti Aher
Assistant Professor, Computer Engineering Department
Fr. Conceicao Rodrigues Institute of Technology
University of Mumbai, Vashi, India
bhakti.aher@fcrit.ac.in

***Abstract*—Disaster communication systems operating in adversarial or high-stress environments have several security risks. In disaster situations, the communication infrastructure that is necessary for emergency coordination is often damaged, preventing victims from asking for help or rescue teams from organising effective response operations. In this paper, we propose ResQMesh, a hybrid infrastructure-independent disaster communication and SOS verification framework that delivers emergency messages without cellular or internet connectivity. The system uses a decentralised multi-node LoRa relay architecture with optional Firebase synchronisation for centralised monitoring when connectivity is available. The multi-layer SOS validation engine reduces false alarms by combining behavioural, environmental, and spatial signals using weighted confidence scoring and spatial correlation analysis. The prototype-scale test further demonstrated the end-to-end latency of between 1.8 and 2.4 s with the 97.8% success rate of message delivery under the controlled conditions.**

***Index Terms*—Disaster Communication, LoRa, ESP32, Emergency Response, SOS Verification, Hybrid Network Architecture, False Alert Detection, Multi-Hop Relay Communication**

I. Introduction

Disasters also often destroy the communication infrastructure for emergency coordination. This makes it impossible for victims to ask for help or for rescue teams to coordinate effective response operations. Large scale emergencies like earthquake, flood, landslide, cyclone etc. often damage the cellular towers, internet backbones and power systems leading to major communication blackouts. The cellular infrastructure collapsed in the 2010 Haiti earthquake, leaving millions of people without a reliable way to contact rescue teams [1]. Similarly, following the Tōhoku earthquake in Japan, studies reported call blockage rates exceeding 90% and underscored the fragility of centralised communication systems in a crisis.

Today's emergency communications have serious disadvantages. GSM and internet based systems depend on the surviving infrastructure, while satellite communication solutions are expensive and impractical for mass civilian deployment. WiFi and Bluetooth-based short-range mesh systems have coverage and scalability limitations in disaster environments.

In this paper, we propose ResQMesh, a hybrid infrastructure-independent disaster communication framework to tackle these challenges. ResQMesh is a hybrid mobile-embedded system that uses an Android app, ESP32 microcontrollers and LoRa radio modules to send SOS messages over long distances without needing cellular or internet connectivity. The system employs a decentralised multi-node LoRa relay architecture with optional Firebase synchronisation for internet connectivity. ResQMesh also features a multi-layer SOS verification engine that combines behavioural, environmental and spatial signals to reduce false alerts and improve emergency prioritisation.

II. Literature Survey

Disaster communication has been extensively studied on different platforms including GSM/SMS systems, satellite

communication, mesh networking, and LoRa-based emergency architectures. Traditional GSM and internet based systems depend on surviving infrastructure and fail badly during disasters, with reported call blocking rate of more than 90% due to damage to towers and congestion of network [1]. Satellite communication systems, such as Iridium and Thuraya, are independent of local infrastructure, but are still too expensive and impractical for widespread civilian deployment [2].

In the field of mesh networking, Ashraf et al. have proposed WiMesh [3], a decentralised communication framework based on WiFi. However, it is limited by a short per-hop range and relatively high power consumption. Alvarez et al. [4] investigated Bluetooth mesh communication using Bluemergency, which provided low power operation but had limited range and increased latency for larger multi-hop networks. LoRa-based communication systems [2], [11] are promising for disaster communication, offering long-range, low-power operation in unlicensed ISM bands. Nevertheless, the majority of current implementations are standalone systems lacking cloud redundancy, intelligent SOS verification, or thorough latency characterisation.

Research on crowd-sourcing in emergency intelligence and probabilistic alert validation [10] shows that spatially correlated emergency signals are superior to isolated alerts in predicting real disasters. These concepts are the basis of the multi-layer SOS verification engine in ResQMesh that integrates behavioural, environmental and spatial analysis for false alert reduction and emergency prioritisation.

Current systems address parts of the disaster communications problem but generally do not integrate infrastructure independence, long-range decentralised communication, hybrid cloud redundancy, and intelligent SOS validation into a single framework. ResQMesh is built to fill this gap.

TABLE I
COMPARISON OF EXISTING DISASTER COMMUNICATION APPROACHES

Work / System	**Method**	**Limitations**	**Outcome**
GSM/SMS Systems	Cellular network	Infrastructure failure during disasters	Functional only under normal connectivity
Satellite Systems	Satellite communication	High cost; specialized hardware	Infrastructure-independent but not civilian-scalable
WiMesh [3]	WiFi mesh networking	Short range; high power consumption	Decentralized local communication
Bluemergency [4]	Bluetooth mesh + IoT	Limited range; high multi-hop latency	Post-disaster BLE communication
LoRa Systems [2]	LoRa communication	No cloud backup or SOS verification	Long-range low-power communication
ResQMesh	Hybrid LoRa + Cloud + Verification	Prototype-scale evaluation	94% false alert accuracy; 97.8% delivery rate

III. PROPOSED SOLUTION

ResQMesh architecture is a three-layer hybrid architecture as shown in Fig. 1 and consists of an Application Layer, Gateway Layer, and Communication Layer. These layers provide disaster communications that are independent of infrastructure with optional cloud connectivity when internet is available.

The Application Layer involves two Android applications and a web dashboard. The Victim Application is written in Kotlin and Jetpack Compose and allows one-tap transmission of an SOS message with GPS coordinates to ESP32 WiFi access points in the vicinity. The Rescue Team Application offers real-time SOS monitoring, offline mapping and responder coordination with Firebase Cloud Messaging (FCM). The Admin Dashboard has been developed using Next.js and React, and supports live SOS visualisation, suspicious-alert confidence indicators as well as integration with Ambee environmental API [5] for weather and flood intelligence.

The Gateway Layer comprises ESP32 microcontrollers [6] and Semtech SX1278 LoRa transceivers. Access Point Nodes offer local WiFi hotspots and receive SOS submissions using lightweight HTTP interfaces, then re-broadcast them as LoRa packets. Hop-count-limited forwarding is used by relay nodes to ensure predictable multi-hop propagation in constrained disaster conditions without the need for dynamic route discovery or routing-table maintenance. Gateway Nodes connect the LoRa relay network to Firebase Firestore whenever the internet is available.

Unlike Bluetooth mesh systems such as Bluemergency [4] which rely on the presence of surrounding participating user devices to maintain relay continuity, ResQMesh employs dedicated ESP32-LoRa relay nodes that are able to communicate over a kilometre scale, regardless of civilian device density. This enhances the communication reliability in disaster environments where the user distribution may be sparse or unstable.

The Communication Layer runs on parallel LoRa and Cloud Channels. The LoRa physical layer uses Chirp Spread Spectrum modulation in the 433 MHz ISM band with configurable spreading factors between SF7 and SF12. The spreading factor SF10 and a bandwidth of 125 kHz with a transmit power of 20 dBm were chosen to strike a compromise between range and latency, with a transmission time of about 220 ms for a 200-byte SOS packet in a range of 5 km. The cloud channel uses Firebase Firestore and FCM for real-time synchronisation and delivery of push notifications when connected.

In the absence of infrastructure, SOS alerts are propagated (multi-hop) via WiFi-connected ESP32 nodes and LoRa relay forwarding to a gateway node or rescue endpoint. In hybrid mode, LoRa and cloud transmissions are simultaneous, and the first successful transmission is taken as received. The dual-path system architecture mitigates single points of failure and enhances the reliability of emergency messages in disaster situations.

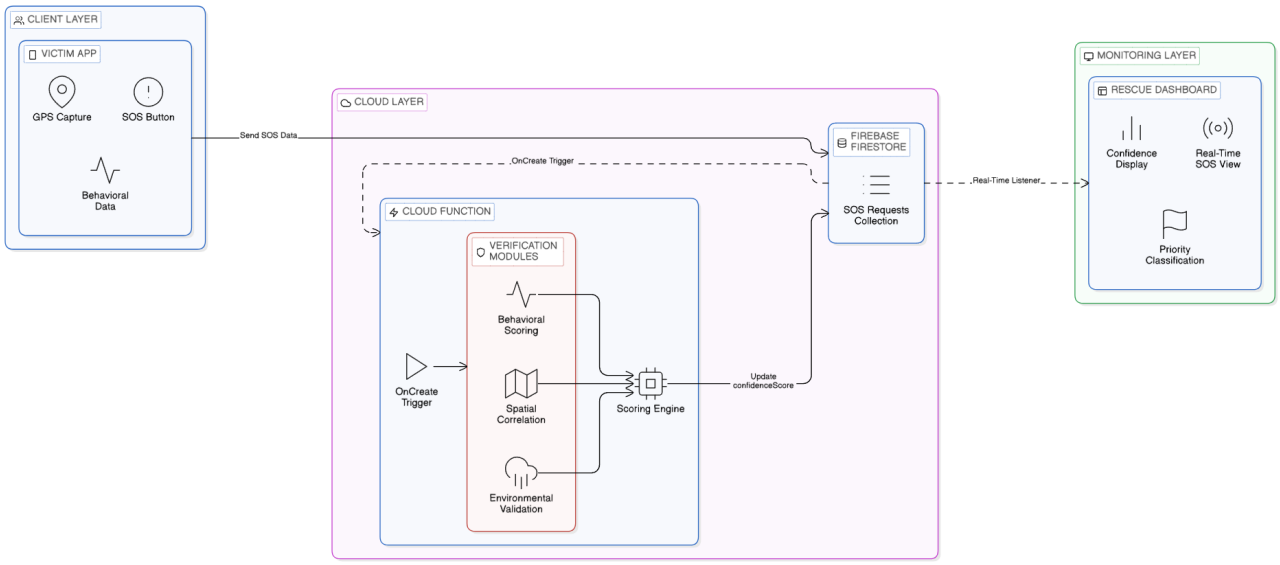

Fig. 1. The ResQMesh hybrid architecture consists of the Application, Gateway, and Communication layers.

IV. Implementation

A. Android Application

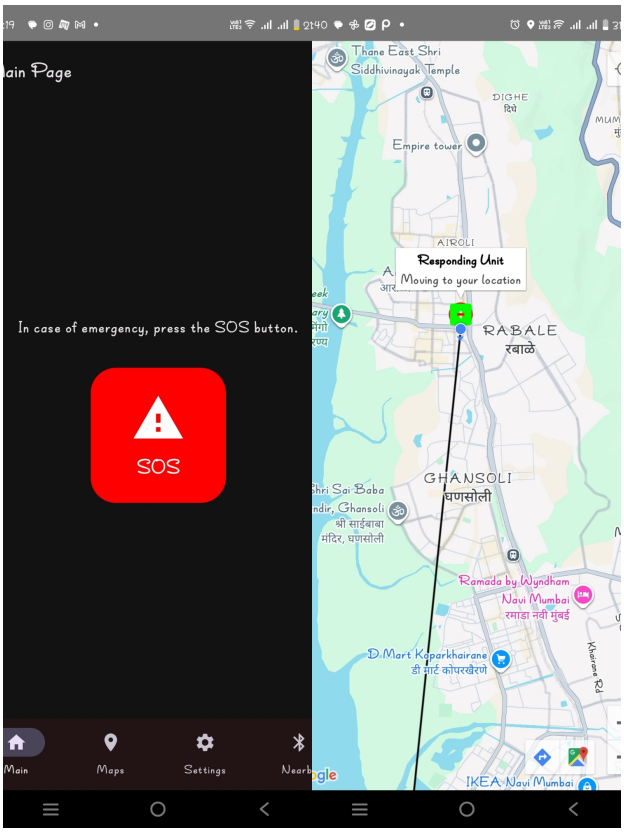

Fig. 2. Victim application interface.

The Victim Application, shown in Fig. 2, enables one-tap SOS transmission by capturing GPS coordinates, attaching a unique device identifier and timestamp, and forwarding the payload to nearby ESP32 access nodes. If no ResQMesh WiFi hotspot is available, alerts are queued locally and retransmitted automatically once connectivity is restored. The application also supports Bluetooth-based peer-to-peer communication between nearby victims and provides delivery acknowledgments after successful SOS transmission.

With internet connection, the application map interface displays the victim locations, nearby SOS alerts and rescue team positions with the help of Firebase synchronisation. The Rescue Team Application allows real-time SOS monitoring and coordination of responders.

B. Admin Dashboard

Figure 3 shows the Admin Dashboard, where rescue coordinators can view SOS alerts, confidence scores and overlays of environmental hazards in real-time. To facilitate prioritisation during disaster response operations, alerts are classified as active, critical or resolved. Integration with Ambee Environmental Intelligence API [5] allows live weather and flood monitoring along with victim locations via Mapbox visualisation and Firebase synchronisation.

C. SOS Validation Logic

The SOS verification engine uses a three-layer scoring framework to reduce false alerts and improve emergency prioritisation. The **Behavioral Layer** (B) looks at repeated SOS activity from the same device within a short time window. The behavioural score is calculated as follows:

$$S = \begin{cases} 0 & \text{if } n \leq 3 \\ \min(100,\ 20(n-3)) & \text{if } n > 3 \end{cases} \quad (1)$$

where n is the number of SOS submissions from the same device in five minutes. The **Environmental Layer** (E) links

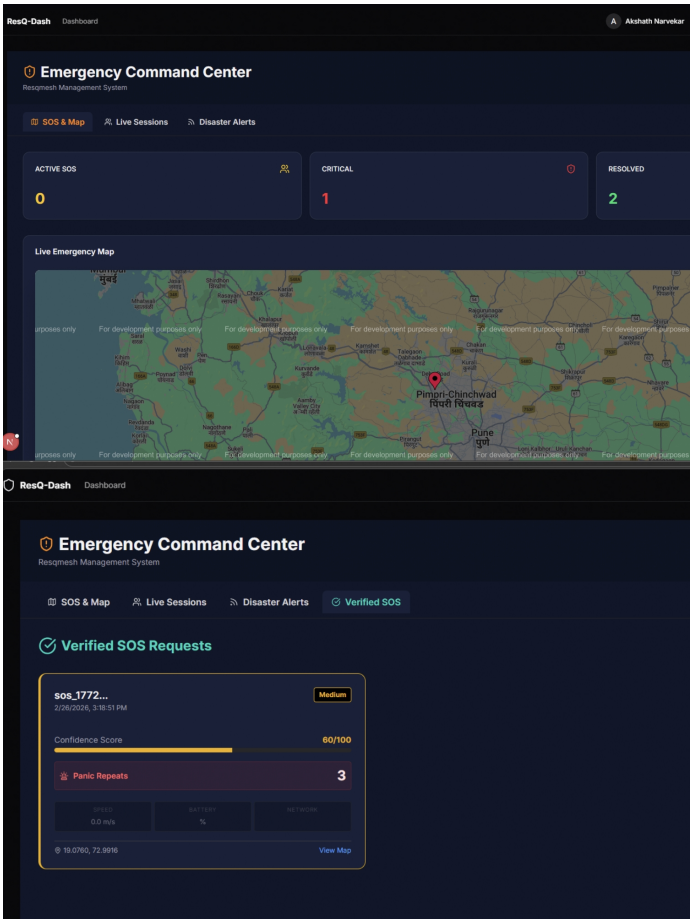

Fig. 3. Admin dashboard for SOS monitoring and disaster visualization.

alerts with weather severity indicators including rainfall intensity, wind speed and active flood alerts obtained from the Ambee API. The **Spatial** Layer (S) checks the spatial clustering of the neighbouring SOS requests, so that correlated alerts within a local area raise the confidence that a true disaster event is happening. The final confidence score is calculated as:

$$C = \alpha B + \beta E + \gamma S \tag{2}$$

where $\alpha + \beta + \gamma = 1$ where the weights are empirically calibrated to $\alpha = 0.10$, $\beta = 0.65$ and $\gamma = 0.25$. Environmental and spatial evidence were more heavily weighted as they were more reliable in the validation experiments. The classification thresholds are:

$$\text{Confidence} = \begin{cases} \text{High} & \text{if } C \geq 70 \\ \text{Medium} & \text{if } 40 \leq C < 70 \\ \text{Low (Suspicious)} & \text{if } C < 40 \end{cases} \tag{3}$$

The final classification is done by Bayesian post-processing:

$$P(\text{Disaster}|\text{Evidence}) = \frac{P(\text{Evidence}|\text{Disaster}) \cdot P(\text{Disaster})}{P(\text{Evidence})} \tag{4}$$

$$P(\text{Fake}) = 1 - P(\text{Disaster}|\text{Evidence}) > 0.6 \tag{5}$$

D. Hardware

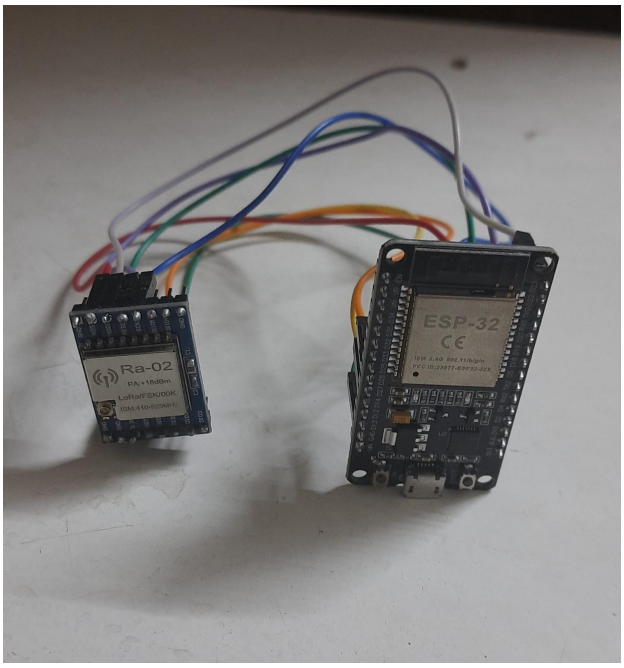

Fig. 4. ESP32-SX1278 LoRa hardware setup.

The ResQMesh hardware backbone is shown in Fig. 4. It is based on ESP32 DevKit v1 boards [6] with Semtech SX1278 LoRa transceivers operating at 433 MHz. The ESP32 nodes act as WiFi access points that collect the SOS submissions from the victim devices and then relay them through the LoRa relay network.

Relay nodes retransmit valid packets received up to a configured relay depth, thus forwarding packets with a hop count limitation. This approach allows for predictable multi-hop propagation without the overhead of dynamic routing. If internet is available, gateway nodes forward incoming LoRa packets to Firebase Firestore.

The system is designed to be suitable for disaster environments that are partially damaged because LoRa Chirp Spread Spectrum modulation is robust to interference and multipath fading. Handling retransmissions and managing duty-cycle at the firmware level enhance communication reliability and regulatory compliance.

V. Results and Discussion

Testing was conducted at Fr. Conceicao Rodrigues Institute of Technology, Vashi, India, using two ESP32+SX1278 LoRa nodes operating at 433 MHz with SF10 and approximately 40 Android client devices distributed across both nodes. Each node was evaluated with up to 20 simultaneously connected devices under repeated SOS transmission conditions in both indoor and outdoor environments.

A. Experimental Methodology

Evaluation was performed using controlled simulations due to the lack of publicly available disaster communication

datasets. A total of 100 scenarios (60 genuine, 40 fabricated) were evaluated using randomized environmental, behavioral, and spatial parameters drawn from Table II. Hardware reliability testing was additionally conducted over six continuous hours with 500 SOS transmissions.

TABLE II
SIMULATION INPUT VARIABLE RANGES

Variable	Range
Rainfall Intensity	0–120 mm/hr
Wind Speed	0–110 km/h
Nearby SOS Count	0–15
Repeated SOS Attempts	1–10
GPS Movement Speed	0–20 km/h
Flood Alert Status	Active / Inactive

B. Multi-Client Stress Testing

Stress testing was performed with up to 20 Android devices connected to a single ESP32 access node under concurrent SOS transmission conditions. No observable packet corruption, duplicate delivery, or routing instability was detected during evaluation.

TABLE III
MULTI-CLIENT STRESS TESTING RESULTS

Metric	Value
Total Client Devices	40
Devices per Node	20
LoRa Nodes	2
Average Latency Under Load	∼2.0–2.5 s
Packet Corruption	None Observed
Routing Instability	None Observed
Delivery Reliability	High

Fig. 5 shows average end-to-end latency under increasing client load. Latency remained below 2.5 s up to 20 devices per node, remaining well within emergency communication requirements.

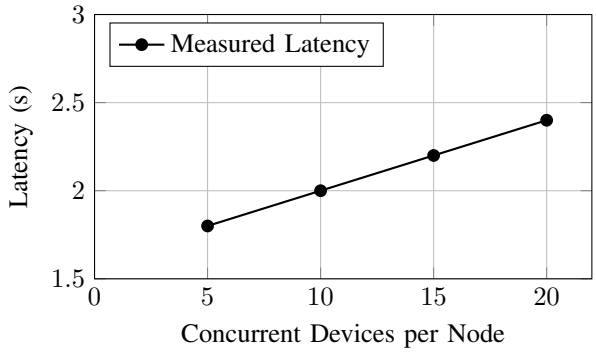

Fig. 5. Average end-to-end latency under increasing client load.

C. Communication Performance

End-to-end latency across all communication paths remained between 1.8 and 2.4 s during testing. Hybrid mode achieved the lowest observed latency by prioritizing the first successfully completed transmission path. Despite raw LoRa packet loss rates of 8–12%, retransmission recovery and hybrid-path redundancy maintained delivery reliability above 95%.

TABLE IV
CLOUD AND LORA COMMUNICATION PERFORMANCE METRICS

Metric	Cloud Mode	LoRa 2-Hop
End-to-End Latency	1.9 ± 0.4 s	2.3 ± 0.5 s
Hybrid Mode Latency	1.8 ± 0.3 s	
SOS Submission Time	0.8 ± 0.2 s	—
FCM Delivery	1.5 ± 0.5 s	—
LoRa Packet Transmission	—	220 ± 15 ms
Multi-hop Delay	—	250 ± 30 ms
Outdoor LoRa Range	2.8 km	
Indoor LoRa Range	450 m	
Packet Loss Rate	—	8–12%
Message Delivery Rate	99.2%	95–98%

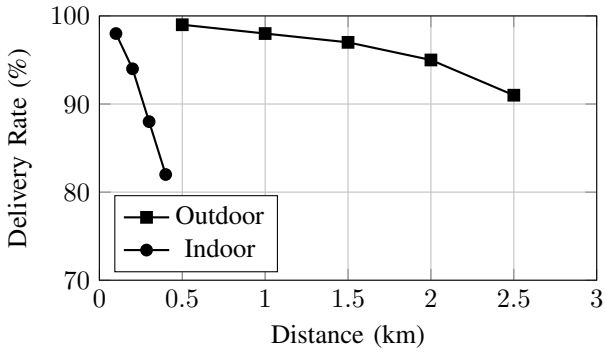

Fig. 6. Packet delivery rate versus communication distance.

Fig. 6 illustrates packet delivery rate versus communication distance for indoor and outdoor environments. Outdoor communication remained above 95% reliability up to approximately 2 km, while indoor performance degraded more rapidly due to wall attenuation.

D. False Alert Detection

Validation results for the SOS verification engine are summarized in Table V.

TABLE V
SOS VALIDATION ENGINE PERFORMANCE

Metric	Value
Overall Accuracy	94%
True Positive Rate	95%
True Negative Rate	92.5%
False Positive Rate	7.5%
False Negative Rate	5.0%
Precision	95%
Recall	92.5%
F1-Score	93.7%

The three-layer confidence model combining behavioral, environmental, and spatial analysis achieved 94% overall clas-

sification accuracy. Environmental and spatial corroboration proved significantly more reliable than behavioral signals alone, motivating the final weight configuration of $\alpha = 0.10$, $\beta = 0.65$, and $\gamma = 0.25$.

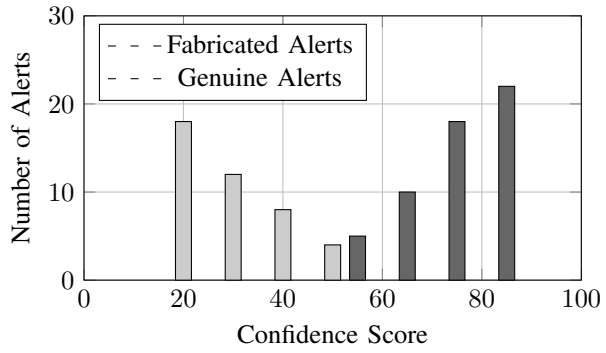

Fig. 7. Confidence score distribution for genuine and fabricated SOS alerts.

E. Comparative Analysis

When SOS verification was turned off, all false alerts were labelled as real. When we turn on the verification engine, the false positive rate goes down to 7.5%. In simulated internet outage conditions, delivery over the cloud-only failed completely, while LoRa-only communication remained at 95–98% delivery reliability. The hybrid mode obtained the best reliability (97.8%) and the least observed latency (1.8 ± 0.3 s). Turning off the environmental validation layer decreased classification accuracy to 81%, further demonstrating the importance of environmental corroboration.

F. System Reliability

During the six hours of continuous stress testing, 500 SOS messages were forwarded, 489 of which were successfully transmitted, for a success rate of 97.8%. During the evaluation, there were no crashes of the ESP32 firmware, memory leaks or gateway failures. The hybrid reliability modelling was consistent with empirical observations in prototype-scale deployment conditions.

VI. Security Considerations

Disaster communication systems operating in adversarial or high-stress environments have several security risks. The current prototype of ResQMesh uses simple protection mechanisms such as unique device identifiers for behavioural tracking, duplicate packet filtering using sequence numbers, and checksums at each relay hop to detect packet corruption.

There are several classes of vulnerabilities left for future work. LoRa-capable devices operating on the same frequency and spreading factor can inject spoofed SOS packets. Duplicate-sequence filtering is only partial for mitigating replay attacks by retransmitting previously captured SOS packets. The existing topology does not cover fake relay-node injection and selective packet dropping too. Furthermore, LoRa jamming on the 433 MHz ISM band is still a physical-layer vulnerability.

Future work will explore lightweight authentication with HMAC-based verification, timestamp-based replay prevention and anomaly detection for routing-layer misbehaviour. Full cryptographic protection is not feasible due to the LoRa payload and duty-cycle restrictions, but selective encryption of SOS packet headers is possible.

VII. Conclusion

ResQMesh proposes a hybrid infrastructure-independent disaster communication framework by integrating multi-node LoRa relay networking, Android-based SOS reporting and multi-layer verification engine with behavioural, environmental and spatial analysis. Experiments show that emergency communication can be made reliable and smart without relying on conventional cellular or internet infrastructure.

Results show that hybrid redundancy between LoRa and cloud channels significantly improves delivery reliability, while multi-layer SOS validation achieves 94% false alert detection accuracy through combined environmental and spatial corroboration. The prototype-scale test further demonstrated the end-to-end latency of between 1.8 and 2.4 s with the 97.8% success rate of message delivery under the controlled conditions.

The current system is still constrained by prototype-scale deployment, synthetic evaluation datasets, and validation beyond two-hop relay operation. Future work will include large-scale field deployment, improved relay coordination, lightweight security mechanisms, machine-learning-assisted SOS verification, and drone-assisted relay expansion for disaster environments where ground infrastructure is not available.

References

[1] United Nations Office for Disaster Risk Reduction, "Global Assessment Report on Disaster Risk Reduction 2022," Geneva, Switzerland, 2022.

[2] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-Range Communications in Unlicensed Bands: The Rising Stars in the IoT and Smart City Scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.

[3] U. Ashraf, A. Khwaja, J. Qadir, S. Avallone, and C. Yuen, "WiMesh: Leveraging Mesh Networking for Disaster Communication in Resource-Constrained Settings," *Wireless Networks*, vol. 27, pp. 2189–2212, 2021.

[4] F. Álvarez, L. Almon, H. Radtki, and M. Hollick, "Bluemergency: Mediating Post-Disaster Communication Systems using the Internet of Things and Bluetooth Mesh," in *Proc. IEEE Global Humanitarian Technology Conference (GHTC)*, 2019.

[5] Ambee Environmental Intelligence Platform, "Disaster and Weather Data API," https://www.getambee.com, 2023.

[6] Espressif Systems, "ESP32 Series Datasheet," Version 4.1, 2023.

[7] Semtech Corporation, "LoRa and LoRaWAN: A Technical Overview," Semtech White Paper, 2022.

[8] Google LLC, "Firebase Documentation: Build and Run Apps," https://firebase.google.com/docs, 2023.

[9] National Disaster Management Authority, India, "National Disaster Management Guidelines: Management of Tsunamis," New Delhi, 2020.

[10] R. Shankar and P. Mehta, "Emergency Communication Systems for Disaster-Prone Regions: A Survey," *Journal of Network and Computer Applications*, vol. 178, p. 103002, 2021.

[11] A. Augustin, J. Yi, T. Clausen, and W. Townsley, "A Study of LoRa: Long Range and Low Power Networks for the Internet of Things," *Sensors*, vol. 16, no. 9, p. 1466, 2016.

[12] T. Kobayashi, "Lessons Learned from the Great East Japan Earthquake: Emergency Communication Failures," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 166–172, 2012.

[13] C. Perkins, *Ad Hoc Networking*, Addison-Wesley, 2001.

[14] Google LLC, “Firebase Cloud Messaging: Architecture Overview and Message Delivery,” in *Firebase Documentation*, Google Developers, https://firebase.google.com/docs/cloud-messaging/concept-options, 2023.
[15] Semtech Corporation, “AN1200.22: LoRa Modulation Basics,” Semtech Application Note, Rev. 2, https://www.semtech.com/uploads/documents/an1200.22.pdf, 2015.