RESEARCH PAPER

Rescue Net – Emergency Response System

Ms. Geetha V[1], Ms. Indhumathi S[2], Ms. Aarthi T[3], Mr. Ajithkumar A[4]
[1,2,3] Assistant Professor, Computer Science and Engineering, KGISL Institute of Technology, Coimbatore, India
[1] Email: geethavmca@gmail.com
[2] Email: indhumathisme@gmail.com
[3] Email: aarthitnila@gmail.com
[4] Assistant Professor, Computer Science and Engineering, JCT College of Engineering and Technology, Coimbatore, India. Email: Ajithkumar.a1627@gmail.com

Received: 2nd Mar, 2026 | **Revised:** 14th Mar, 2026 | **Accepted:** 4th Apr, 2026 | **Available Online:** 20th Apr, 2026

***Abstract*—**Due to ineffective communication between victims and rescue services, delayed response times are a major prob- lem in disaster management, frequently leading to fatalities and increased damage. Conventional systems are not scalable during periods of high load, depend on manual alerts, and lack prioritization. RescueNet is a Django-based multi-agent platform that streamlines emergency workflows by integrating GSM technology for call alerts, role-based dashboards, and real- time tracking. Its modular design makes emergency response quicker, more intelligent, and more effective in a variety of public settings by increasing operational speed, accessibility, and reducing confusion.

***Index Terms*—**Emergency Response, Multi-Agent System, Of- fline Support, Alert System, Public Safety, Minimal-Click Design, Disaster Management

How to cite this article: Geetha V, Indhumathi S, Aarthi T, Ajithkumar A. Rescue Net – Emergency Response System. Int J Drug Deliv Technol. 2026;16(34s):468-474. DOI: 10.25258/ijddt.16.34s.63

I. Introduction

Delays in emergency response continue to be a major problem globally, frequently resulting in a rise in casualties and ineffective use of resources. The majority of current systems relies on manual procedures, physical infrastructure, and dispersed communication, all of which can be expensive, slow, and uncoordinated in real-time situations.

In order to get past these obstacles, we present RescueNet, a multi-agent emergency response system powered by Django that is designed for important situations like natural disasters, medical crises, and accidents. The platform ensures quick information flow by integrating GSM module support for call- based alerting.

RescueNet places a strong emphasis on alert tracking, real-time communication, and scalable deployment in public, institutional, and residential settings. It increases operational efficiency and reduces human error through automated work- flows and an intuitive interface. RescueNet seeks to modernize emergency management and promote safer, smarter communi- ties worldwide through intelligent, real-time coordination by increasing accessibility and decreasing response latency.

Related Work

For many years, emergency response systems have been based on centralized models that frequently rely on manual communication and paper-based workflows, which can cause delays and poor management in urgent situations. Conven- tional disaster management platforms and traditional systems, such as 911, offer basic dispatch and routing capabilities, but they lack real-time intelligence and modularity—two things that are critical for a high-efficiency response.

In order to improve response time and eliminate bottle- necks, multi-agent systems have been investigated for their ability to decentralize decision-making and divide tasks among autonomous units. However, these implementations are fre- quently theoretical or limited to particular domains, such as simulations of fire evacuation or smart traffic. Although there are still few real-world, affordable

implementations, existing research also emphasizes the advantages of using priority- based dashboards and modular communication layers.

In places with limited infrastructure, GSM-based alerting systems have become popular because they allow for basic communication without relying on the internet. Still unex- plored, though, is integration with role-specific dashboards and structured agent-based routing. RescueNet expands on this environment by integrating a multi-agent priority handling model, GSM-enabled alerts, and a Django-based architecture. In contrast to previous systems, it provides modular scalability, real-time alert categorization, and user-friendly dashboards to expedite rescue coordination in dynamic, real-world settings.

II. Problem Statement

Medical emergencies and natural disasters are examples of emergency situations that require quick, well-coordinated, and astute reactions in order to reduce damage and fatalities. However, traditional emergency response systems frequently have serious drawbacks like slow response times, ineffective channels of communication, and a lack of task allocation

based on priority. These flaws can seriously impede prompt action, particularly in places with limited infrastructure or high population densities.

Traditional systems mainly rely on centralized dispatching and manual alert processing, which can lead to misunderstand- ings, misallocation of resources, and miscommunication dur- ing periods of high demand. A unified platform that combines modular task routing, multi-agent coordination, and real-time priority handling is frequently absent from current solutions, despite the fact that some of them include GPS tracking or simple mobile notifications.

A real-time, scalable, and easily accessible emergency re- sponse system that offers effective communication, dynamic prioritization, and smooth task distribution is desperately needed. In order to close this gap and enhance rescue coordi- nation, RescueNet integrates multi-agent architecture, Django- based infrastructure, and GSM-based alert mechanisms. Res-cueNet's role-specific dashboards, alert tracking, and flexible workflows improve efficiency and accountability while guar- anteeing quicker and more intelligent emergency responses.

III. Proposed System and Methodology

The RescueNet system, which is intended to expedite emergency response activities during critical incidents like medical emergencies, accidents, and disasters, is described in this section along with its architecture, essential parts, and working workflow. With the goal of reducing response times and increasing rescue efficiency, RescueNet provides a real-time, modular, and priority-based solution for improving communication between victims and rescue teams.

A. System Architecture

Key elements of the RescueNet system architecture include the Rescue Dashboard, the Arduino Interface, the GSM Com- munication Module, the Natural Language Processing (NLP) Engine, the Graphical User Interface (GUI), the Backend Processing Module (created with Django), and the Priority Assignment Unit. Users' emergency input, such as the incident type and location, is gathered by the GUI. The backend receives this data and stores it in CSV format for use in future model training as well as real-time processing.

The NLP Engine automatically assigns a priority level to the message after analyzing the input to determine the situation's urgency and context. The Arduino board receives this assigned priority and serves as a mediator to forward it to the GSM Module. Call-based alert transmission is made possible by the GSM Module, guaranteeing smooth communication even in places with poor connectivity. All data is simultaneously locally logged by the system for accountability and future analysis.

B. Smart Priority Assignment System

The RescueNet system interprets user inputs, finds key-words related to emergencies, and classifies the type of emergency using a clever priority assignment mechanism driven by Natural Language Processing (NLP). Following

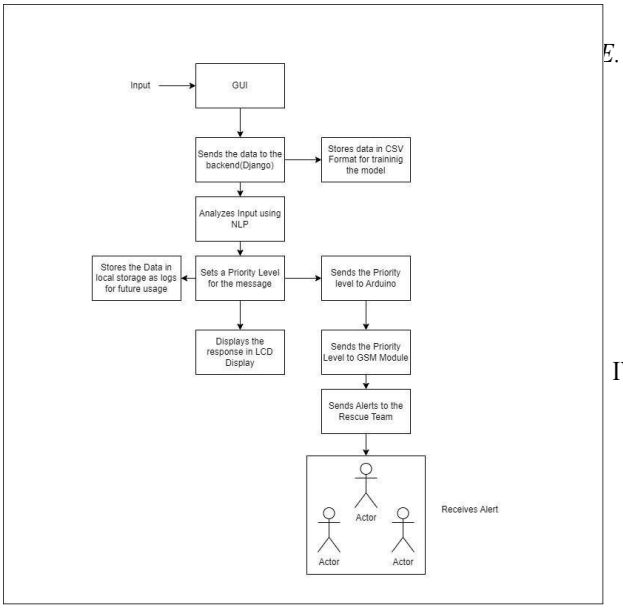

Fig. 1. System Architecture of RescueNet

analysis of the input, the system designates a priority level—high, medium, or low—according to the gravity and urgency of the circumstance. Rescue crews can maximize their response time and resource allocation by ensuring that critical emergencies are addressed first thanks to this automatic classification.

C. Communication and Alert Transmission

RescueNet uses a transmission system based on Arduino and GSM to enable efficient communication with rescue teams. Following the setting of the priority level, the data is transmitted to an Arduino, which forwards it to a GSM module for the creation of an SMS alert in real time. Regardless of location or network constraints, these alerts are sent to the appropriate rescue personnel, guaranteeing prompt notification. Delays in rescue and support operations are reduced thanks to this dependable two-stage communication system, which ensures that vital alerts get to responders promptly.

D. Data Storage and Training Module

uses both structured and local storage techniques to guarantee reliable data management. Emergency infor- mation is instantly entered into a local database for easy access and future use. Furthermore, a CSV data logger keeps track of a formatted dataset that can be utilized to gradually train and enhance the NLP models.

E. Responder Interface and Monitoring Dashboardk

Responders are encouraged by the RescueNet system to keep an eye on alerts and emergency trends over time, modify- ing their tactics for the best possible rescue results. Responders

can use the system to record feedback if they experience de- lays, a lack of resources, or problems with coordination. This information is then stored for future analysis and operational enhancements.

IV. Implementation and Results

A functional prototype was created and tested in a variety of emergency situations, such as medical, fire, and accident cases, in order to validate the RescueNet system. The system makes it easier to categorize, report, and route emergencies to respon- ders in real time. To guarantee prompt and well-coordinated rescue responses, it integrates GSM module communication, centralized dashboard monitoring, and alert prioritization.

A. Development Stack

A layered architecture was used in the development of the RescueNet application, which prioritized user-friendly respon- der interfaces and real-time alert processing. For call-based alerts, GSM modules were integrated, guaranteeing smooth emergency reporting even in the absence of internet access. To power the backend Django is used in order to manage and handle data flow effectively.

- **Front-end**: HTML, CSS (for dashboard and responder interface)
- **Communication Module**: GSM Module SIM800L (for call alerts)
- **Backend**: Django Framework (for database handling and logic processing)
- **Database**: MySQL (for storing alert logs and status updates)
- **Tools**: Arduino IDE (for GSM programming), Postman (for API testing)

B. Module Implementation and Testing

The system was divided into three main components: The Responder Dashboard, GSM Call Module, and Emergency Alert Generation

Through a straightforward interface, victims could send comprehensive alerts in the Emergency Alert Module, including type, location, and priority.

The GSM Call Module provided voice alerts even in the event of an internet outage by automatically dialing designated emergency numbers.

The Responder Dashboard displayed real-time status up- dates and arranged incoming alerts by priority.

C. Performance and Latency Testing

The responsiveness and dependability of the system were evaluated in various network scenarios. Among the important performance indicators noted were:

- 5–8 seconds are needed to initiate a GSM call.
- Dashboard Update Latency: less than one second
- Alert Categorization Accuracy: 95

D. Session Storage and Report Generation Time:¡1s*User Evaluation and Feedback*

Volunteers for emergency response were asked to engage with the system in a variety of simulated situations. Using the victim interface, they sent out emergency alerts, and the dashboard and GSM systems were used to track the responses. Results: During drills, responders reported better coordination, quicker decision-making, and quicker identification and
Classification of emergencies.

Session Logs: Timestamps, emergency type, location infor- mation, and action status were recorded for each alert event for later review.

System Analysis: When compared to manual reporting, system analysis showed an average response time reduction of 30

E. Summary

Real-time emergency reporting, GSM-based communica- tion, and effective priority-based alert management were all successfully demonstrated by the RescueNet prototype. It bridged the gap between early alert generation and prompt rescue operations, proving to be a scalable, economical, and highly responsive solution for contemporary emergency response systems.

V. Results and Discussion

The evaluation results of the Priority Dashboard/Communication System, which was created to improve communication and expedite message prioritization, are shown in this section. The system's functional performance, user interface effectiveness, and data accuracy in managing and prioritizing messages in real-time are the main areas of evaluation.

A. System Functionality Evaluation

A controlled environment with simulated real-time message scenarios was used to assess the Priority Dashboard. The following results were shown by the system:
The system demonstrated:

- High-priority messages were effectively categorized and prioritized by the dashboard for prompt attention.
- Quick priority updates and smooth interaction.
- The dashboard performed consistently and error-free.
- The system processed messages with the fewest possible delays.

The system flow includes:

- Using the user interface to record incoming messages.
- Message processing according to user input and priority settings.
- Prioritized messages are shown on the dashboard in real time.
- AI-powered monitoring of performance indicators

B. Personalization and Feedback

The Priority Dashboard places a strong emphasis on cus- tomizable settings and user-centric design:

- Message prioritization levels and notification preferences can be customized by users to suit their own require- ments.
- For continuity and data-driven performance analysis, ses- sion logs are kept for each interaction.
- Dynamically modifying dashboard settings to fit user workflows guarantees the communication management.

The system promotes:

- enhanced efficiency in handling messages by prioritizing
- Consistent feedback and progress tracking

C. Progress Tracking

A dedicated AI module tracks user progress across sessions:

- Performance is recorded daily, and graphs are used to show improvements.
- Using session history, the system suggests the best mes- sage prioritization settings.
- Users are able to track the results of communication management

These features can efficiently monitor the results of commu- nication management and make data-driven modifications.

TABLE I
Comparison of Proposed RescueNet System with Existing Emergency Response Tools

Feature	Proposed System (RescueNet)	Basic Emergency Apps	Traditional Emergency Response
Platform	Mobile/Web (Flutter)	Mobile/Web	Physical dispatch centers
Location Tracking	Yes (Real-time updates)	Limited	Manual reports
Alert Notification	Yes (Automatic and manual)	Manual only	Manual
Customization	Yes (User-configurable alerts)	No	Limited
Progress Tracking	Session-based monitoring	No	Manual
Data Visualization	Daily incident summaries	No	No
Rescue Continuity	Yes (Session auto-logging)	No	Dispatcher-dependent
Accessibility	High (Remote access)	Medium	Low (On-site reporting)
Scalability	High (Modular system)	Medium	Low

D. Discussion

The suggested RescueNet system successfully tackles im- portant emergency communication issues like centralized mon- itoring, real-time routing, and alert prioritization. RescueNet offers an organized, technologically advanced platform that improves emergency response coordination through automatedAlert management and visual dashboards, in contrast to con- ventional systems that depend on manual communication or irregular information flow.

The system guarantees reliable communication even in low-connectivity settings by integrating real-time responder dashboards and GSM-based voice alerting. Responders can act more quickly when emergencies are categorized according to their type and urgency, reducing delays brought on by human error or information bottlenecks in high-stress scenarios.

RescueNet distinguishes itself from simple emergency re- porting systems by providing a single interface that supports digital and call-based alerts, along with visual cues for loca- tion, criticality level, and response status. Teams can prioritize alerts according to their severity and the resources at their disposal thanks to the dashboard's organization, which makes decision-making easier.

VI. Security Considerations

The RescueNet system relies heavily on security, especially when it comes to responder privacy, alert authenticity, and emergency data integrity. The main steps taken to protect private communications and guarantee system reliability are described in this section.

A. Tamper Resistance

A secure cloud database with role-based permissions houses all alert data, including responder actions, priority tags, and user-generated reports. To maintain system integrity during real-time operations, alerts can only be submitted or modified by verified devices or dashboards.

B. Authentication and Alert Validation

Every alert sent via digital interface or GSM is checked against a distinct source identifier. Only authorized personnel are able to view or respond to classified alerts thanks to device- level authentication, which controls responder access.

C. Replay Attack Prevention

Exercises are tracked daily using a currentDay field stored per user. This ensures that patients progress sequentially through the therapy plan and cannot reuse or skip exercises arbitrarily. Attempts to replay or skip sessions are invalidated by backend verification logic.

D. Data Privacy and Network Control

RescueNet avoids replaying previous alerts or duplicate emergency reports by using timestamp-based verification and distinct session tokens. This keeps confusion at bay during mass-response events and guarantees that every emergency input is

handled differently.

E. Attack Surface Mitigation

RescueNet defends against the following major threats:

- **Man-in-the-middle (MITM)**: To prevent interception or eavesdropping, all communication between devices and the server is encrypted.
- **Unauthorized Access:** Emergency alerts can only be received, processed, or responded to by dashboards that have been verified and registered.
- **Data Tampering:** To guarantee their integrity and stop any unwanted changes during transmission, alerts are verified at both the local and central systems.

RescueNet is a safe and dependable communication plat- form for emergency response coordination because of these features.

VII. Limitations and Future Work

RescueNet has a lot of potential to revolutionize emergency communication, but there are still room for improvement and expansion due to certain operational and technical limitations.

A. System Limitations

- **Network Dependency:** A dependable network connection is necessary for all of RescueNet's essential operations, which depend on cloud-based data handling and communication.
- **Basic Alert Categorization:** The existing system uses preset rules to categorize emergencies, which might miss complex situations or changing field conditions that call for quick decisions.
- **Limited Cross-Agency Coordination:** Although respon- ders can view alerts on their dashboard, the system does not yet allow real-time coordination between various rescue units or outside agencies.
- **No Offline Data Logging:** Alerts from offline zones cannot be stored or forwarded by the system. Critical incidents may not be recorded or sent until a connection is restored in the event of an outage.

B. Future Work

Future developments will concentrate on the following to improve RescueNet's performance and impact:

- **Offline Alert Capture and Sync:** To store data during outages and have it automatically sync with the central dashboard when connectivity is restored, install offline emergency logging at local units.
- **Integrated GPS Tracking:** During crucial rescue opera- tions, outfit responder units with GPS tracking for better coordination, real-time location, and route optimization.
- **End-to-End Messaging System:** For updates and communication between command centers and field agents that go beyond alerts, put in place a safe, two-way messaging system.
- **Multilingual and Icon-Based Interface:** To guarantee accessibility for users in crisis areas with varying lin- guistic and literacy backgrounds, include visual icons and support for regional languages..
- The goal of these enhancements is to make RescueNet a more dependable, field-ready platform for effective emergency response and coordinated disaster management.

VIII. Conclusion

RescueNet, a safe, quick, and coordination-oriented emer- gency communication system, is introduced in this paper. The system guarantees organized prioritization and prompt infor- mation delivery in emergency situations by integrating GSM- triggered alerts, centralized dashboards, and on-site display modules.

RescueNet integrates real-time status tracking and alert validation, enabling responder teams to receive, visualize, and respond to emergency alerts based on dynamic urgency levels. While the cloud dashboard centralizes command visibility, local LCD displays improve field clarity.

RescueNet connects frontline responders and control units through a scalable architecture, GPS support, and offline fallback capabilities, enhancing the intelligence, speed, and accountability of rescue operations.

Acknowledgment

The authors express their gratitude to the KGISL Institute of Technology's Department of Computer Science and En- gineering for their invaluable assistance and direction during RescueNet's development. A special thank you to Ms. Geetha V, our project mentor and assistant professor, whose unwaver- ing support, technical advice, and oversight helped to shape the course of this work.

In addition, we would like to thank our volunteers and peers for their input, testing, and simulations, all of which were essential to the successful development and improvement of the RescueNet emergency communication system.

References

[1] D. K. Sharma and R. Raj, ―Smart emergency response system using IoT and cloud,‖ *International Journal of Engineering Research and Applications*,

vol. 10, no. 4, pp. 12–18, 2020.

[2] A. Kumar and V. Gupta, ―A GSM-based alert system for emergency re- sponse using Arduino and SIM800L,‖ *Journal of Emerging Technologies and Innovative Research*, vol. 7, no. 6, pp. 1021–1026, 2021.

[3] M. Singh, P. Verma, and N. Jain, ―Design and implementation of smart emergency response system,‖ in *Proceedings of the IEEE International Conference on Computing, Communication and Automation*, pp. 645–649, 2019.

[4] R. Mishra and S. K. Dubey, ―Priority-based alert system using GSM mod- ule in emergency services,‖ *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 9, no. 1, pp. 88–91, 2020.

[5] S. Ahmed and T. Hussain, ―Agent-based modeling for emergency man- agement systems,‖ *International Journal of Computer Applications*, vol. 157, no. 5, pp. 17–22, 2017.

[6] L. Zhang, X. Li, and Y. Wang, ―Development of an intelligent emer- gency management system using IoT and big data,‖ *Journal of Systems Engineering and Electronics*, vol. 30, no. 3, pp. 577–584, 2021.

[7] J. Patel, A. Soni, and M. Mishra, ―GSM-based emergency communication system for disaster management,‖ *International Journal of Disaster Risk Reduction*, vol. 42, pp. 205–211, 2019.

[8] T. H. Kim, H. S. Lee, and S. Y. Cho, ―Smartphone-based emergency alert system using wireless sensor networks,‖ in *Proceedings of the IEEE International Conference on Communications*, pp. 679-683, 2020.