2025 IEEE 4th International Conference for Advancement in Technology (ICONAT)
Goa, India. Sep 19-21, 2025

LLM and MCP Enabled AI Agent for Autonomous Multi-Lingual Alerts for Coastal Communities

Sruthy Anand
Center for Wireless Networks & Applications (WNA)
Amrita Vishwa Vidyapeetham
Amritapuri, India
sruthyanand@am.amrita.edu

A M Abhishek Sai
Software Development Engineer
Fridayy AI
Noida, India
abhisheksai1456@gmail.com

Adithya D Varma
Department of Computer Science and Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Amritapuri, India
amenu4aie21104@am.students.amrita.edu

Sethuraman N. Rao
Center for Wireless Networks & Applications (WNA)
Amrita Vishwa Vidyapeetham
Amritapuri, India
sethuramanrao@am.amrita.edu

Alberto Montresor
Department of Information Engineering and Computer Science
University of Trento
Trento, Italy
alberto.montresor@unitn.it

Maneesha Vinodini Ramesh
Center for Wireless Networks & Applications (WNA)
Amrita Vishwa Vidyapeetham
Amritapuri, India
maneesha@amrita.edu

***Abstract*—Intelligent and autonomous systems are essential for ensuring effective data dissemination in disaster management, particularly for coastal communities and fishermen. Autonomous systems that can make decisions independently are particularly valuable in responding to the dynamic and rapidly changing real-life situations of fishers as they are more prone to natural disasters during their fishing trips. The recent advancements in Large Language Models (LLMs) have made it possible to envision human-like cognitive systems capable of reasoning. In this paper, the authors focus on designing an autonomous AI system, which is capable of alerting fishermen on various imminent dangers in their preferred language. The system is able to fetch information from multiple sources, analyze the collected data, and make alert decisions based on its reasoning capabilities. The system follows a client-server architecture with LLM-enabled agents and the Model Context Protocol (MCP) which allows the LLM to access multiple tools and API to perform its assigned task. The system prototype is implemented using web scraping, prompting, and multi-channel processing, which automate information retrieval and intelligent analysis of existing warning messages from government websites, and generate customized alerts that are both contextual and user-friendly. The system is designed to operate hands-free and receive emergency alerts as voice alerts. The prototype implementation shows promising results, as the system can easily categorize and send multilingual safety voice alerts.**

***Index Terms*—Voice Alerts, Autonomous system, AI Agents, Model Context Protocol, Disaster Alerts, Resilience.**

I. Introduction

A disaster is an unexpected, sudden event that can lead to the loss of human lives and can seriously disrupt the functioning of a community [1]. Its consequences often exceed the community's ability to cope using its own resources. Hence, disasters often cause mental health problems such as anxiety, stress, and other psychological symptoms [1]. One of the largest natural disasters, the Indian Ocean Tsunami on 26 December 2004, killed 225,000 people and also displaced millions [2]. These coastal communities are often affected by cyclones and other emergencies due to high tides, coastal flooding, and similar factors. The Indian coastal communities rely-largely on fishing as their main occupation [3] and face a serious lack of early warning during emergency situations [4]. Such disasters need clear and detailed communication and a timely response to save lives and prevent loss [5]. During an emergency situation, often the time between the reception of warning time and the time required to take the necessary action is short. Hence, the community needs to be prepared to respond to such events within a short span of time. Many existing data dissemination patterns follow a hierarchy, but hierarchical broadcast-based dissemination is often not effective as the perception of the message by the user largely affects their actions [6]. Hence, the disaster warning information needs to be very clear and specific with actionable information and it should be directed to the community.

Since fishing is the primary occupation of coastal communities, fishermen often spend several days at sea with limited opportunities for communication with the outside world. This lack of connectivity further increases their vulnerability to disasters and emergencies, putting their lives and livelihoods at risk [4]. Hence, autonomous and intelligent warning and alert dissemination mechanisms are essential for coastal fishing communities. The novel OceanNet Digital Ecosystem (ODE), designed using the OceanNet infrastructure and the mobile application [7], also has limitations in providing intelligent and autonomous alerts, warnings, and assistance messages to fishermen at sea. There are multiple challenges associated with the effective dissemination of warning messages. The first is limited connectivity due to the dynamic nature of the network; the second is the lack of autonomous, intelligent security applications capable of generating effective warning messages without multi-level human intervention, which often causes delays in delivering critical and timely information. This paper addresses the second challenge by proposing a system capable of intelligently classifying, compiling, and delivering customized, context-aware information such as warning messages by incorporating the latest AI technologies.

AI agent-enabled autonomous entities are becoming increasingly popular, as they can independently perceive their environment, make decisions, and take actions without external intervention. They are capable of collecting information about their surroundings through various input mechanisms and using it to determine appropriate actions based on their goals [8], [9]. Most recently, LLM-enabled AI agents have gained popularity due to their enhanced reasoning and generalization capabilities across a wide range of applications–from general to specific–as they offer extensive opportunities for exploration [8]. These agents, built on large language models such as GPT-4, combine the strengths of both LLMs and autonomous agents. They leverage LLMs to support cognitive and strategic processes [8]. However, LLMs are inherently designed to operate based on the prompts they receive and, therefore, cannot directly utilize external tools for advanced reasoning. With the recent development of the Model Context Protocol (MCP), LLMs can now access external tools, significantly enhancing their ability to perform complex tasks. In this paper, to support this functionality, we introduce an autonomous intelligent decision module equipped with speech synthesis (text-to-speech). This module generates spoken alerts from customized, context-aware texts that include severity levels determined by the system. This speech generation plays a crucial role in assistive technologies aimed at enhancing human safety and well-being.

The rest of the paper is organized as follows. Section II presents the literature review and Section III presents the need for intelligent automatic alert systems for coastal communities. Section IV architecture and components of the system. Section V presents implementation and results, Section VI presents the limitations of current study and future research directions and Section VII presents the conclusion.

II. Literature Review

Most fishermen lack digital literacy, and in many coastal communities across India, the fishing population exhibits varying levels of digital skills [4]. Delays in the return of fishers from sea often cause concern for their family members [3]. These concerns are justified, as fishing vessels may capsize [4], or fishers may unknowingly cross maritime boundaries, which are often difficult to identify and can lead to life-threatening situations. In some cases, fishermen are kidnapped and their boats confiscated [10]. To address these issues, researchers have proposed fishing vessel tracking systems to monitor the real-time location of vessels. Accurately tracking their position can help warn fishermen when they are nearing boundary limits, and this information, when relayed to a control center, can play a crucial role in safeguarding their lives [11].

During disasters, evacuation alerts are typically sent to personal mobile devices. Some researchers have explored the use of humanoid agents to deliver voice-based evacuation alerts [12], as this approach increases the directness and personal relevance of the message by addressing individual users. Existing studies [13] have examined how people interpret tweets and Wireless Emergency Alerts (WEAs) received on mobile devices, particularly in response to unfamiliar hazards. Recently, the effectiveness of voice alerts has gained increasing attention across various alert applications [14]. Voice messages are often more effective than text in prompting people to take appropriate actions to avoid danger.

As AI continues to enhance personalization [15], and AI-driven algorithms support data-driven decision-making and improved efficiency [16], the capacity of generative AI tools to produce human-like language has made them increasingly popular [17]. According to a recent study [18], LLM-based multi-agent systems (MAS) are considered a promising pathway toward achieving artificial general intelligence, as they demonstrate exceptional reasoning and planning capabilities. They proposed a unified agent framework consisting of five essential models: agent profile (characteristics such as how well-suited the agent is for its assigned function), perception (how the agent perceives the environment to acquire knowledge and experience), self-action (how it stores information and performs reasoning), mutual interaction (how it communicates with other entities), and evolution (how it enhances its intelligence over time). Each agent profile should be designed to accurately represent these roles within the environment in which it is intended to operate.

III. Need for AI Enabled Automatic Alert System for Coastal Communities

Approximately 0.1% of global deaths result from natural disasters [1]. Recent climate changes have increased both the frequency and severity of such events. Coastal communities are particularly vulnerable, facing repeated exposure to hazards such as cyclones, tsunamis, severe weather, and high tides. Frequent cyclones along the Indian West Coast have led to loss of life and significant disruption to livelihoods [19]. A community's ability to respond to and recover from such unexpected shocks is a key determinant of its inherent resilience [20].

Current disaster alert methods are often not personalized or tailored to individual users or communities, and recipients may lack clarity about the actions they should take upon receiving a warning message. Additionally, delays in alert delivery–caused by the hierarchical structure of information dissemination in disaster management–can hinder timely responses. These delays can be reduced by providing real-time alerts directly to end users. Therefore, it is essential to develop systems and applications capable of monitoring environmental conditions and integrating data from alert dissemination platforms such as INCOIS, IMD, and other relevant sources. These systems should generate context-aware warning messages and

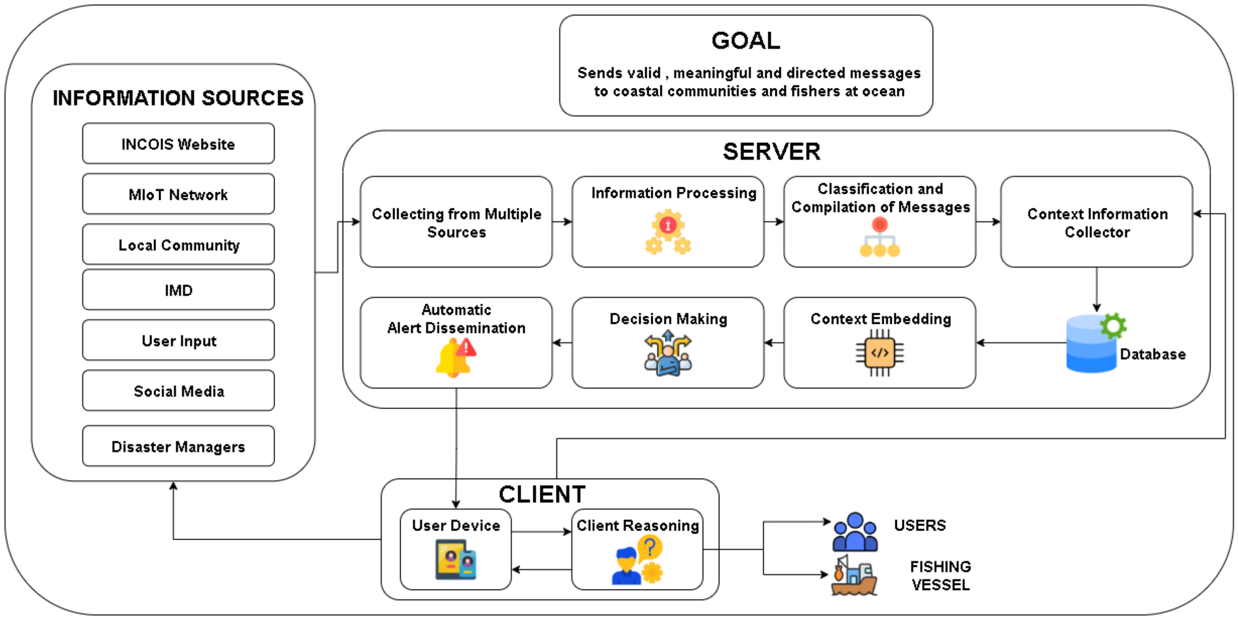

Fig. 1. LLM Enabled Intelligent Alert Dissemination Agentic Reasoning System Pipeline

actionable instructions based on local needs. Furthermore, they must be able to extract, verify, and interpret information from government agency websites to ensure the alerts provided to communities are both accurate and actionable.

Alerts for fishermen and coastal communities are frequently published on the INCOIS and IMD websites. However, these communities often do not check such websites regularly; instead, they rely on peer communication or general broadcast messages. Therefore, there is a critical need for an effective solution that delivers individualized messages to fishermen and coastal residents, clearly indicating the severity of each alert and providing actionable guidance–whether it involves taking immediate action or remaining alert for further updates. Misinformation, including fake, exaggerated, or contradictory messages circulated through instant messaging and social media, often leads to panic. In such cases, people may struggle to accurately interpret official warnings, especially without prior experience [6]. To address this, involving selected community members and delivering properly customized, automated messages in each coastal area could significantly improve disaster preparedness and enhance community resilience.

The proposed LLM-enabled intelligent system is capable of selecting information sources, analyzing data from multiple channels, and generating customized alert messages that include severity tags and recommended actions. This approach aims to provide communities with the information they need to enhance safety and resilience. The method requires context engineering, which involves supplying the system with relevant information, instructions, tools, and prompts to execute its designated functions by leveraging internal memory and knowledge of the target community's environment. In general, an agent is any entity capable of perceiving its environment and autonomously performing tasks across diverse settings, drawing on past experience and knowledge to make decisions aligned with predefined objectives [8]. For LLM-enabled agents, synthetic prompting is often essential to support advanced reasoning. This technique improves the model's decision-making by guiding it with carefully designed example prompts, which in turn allow the model to generate additional examples and further refine its reasoning capabilities [21]. The following section outlines the architecture, components, and process flow of the proposed intelligent system.

IV. System Architecture of LLM and MCP Enabled Intelligent Agentic System for Alerts

For the intelligent system to operate effectively at sea, we assume it can function over any available network, such as cellular or OceanNet, to transmit alert messages. However, networks like OceanNet present greater challenges in terms of reliability due to their self-organizing nature. To ensure the dissemination of alerts to fishing vessels operating in deeper ocean areas, the system must be able to adapt to dynamic network conditions and intelligently determine when and what messages should be sent to users. Traditional systems are typically designed to perform predefined tasks without adapting to changes in the environment. However, with recent advances in AI, programs have become increasingly dynamic, capable of learning from their surroundings and adjusting their actions accordingly. By perceiving the environment and applying rule-based intelligence, an intelligent system can proactively make decisions in pursuit of its defined goals. Such a system must also be designed to be delay-tolerant, meaning messages should be stored locally when connectivity is unavailable. The system must be capable of learning from network conditions and transferring stored messages from the local buffer to another device or directly to the recipient once connectivity is restored.

The proposed LLM-enabled intelligent alert system has to have autonomous capabilities without the intervention to work effectively and hence the dynamic decision module capable of collecting information from multiple sources and classification and reasoning is necessary. The various components of the proposed LLM-enabled AI agent reasoning and alert dissemination is shown in the Fig. 1. The system has a specific goal that serves as its objective and receives information from multiple sources, which provide both information and context. The major components of the system are as given below:

- **Goal:** Any autonomous system must have a clearly defined goal that serves as its primary objective. This goal guides the system in processing inputs necessary to achieve the intended outcomes. One of the key characteristics of an intelligent system is its ability to pose problems to itself, enabling it to make effective use of the information available. It should also be capable of formulating new questions and reasoning based on them. Just as humans internally ask questions to analyze and understand context, an intelligent system must possess the ability to generate and reflect on its own queries.
- **Information Retrieval:** Information from multiple sources can be integrated into the system. Information retrieval or acquisition is vital for an intelligent system enabled with an LLM agent, as it allows the agent to analyze context–i.e., to perceive external environmental conditions–and make appropriate decisions. Retrieved data is stored internally, often in databases or JSON formats, to facilitate efficient processing. This structured information is then used to support autonomous decision-making and response generation.
- **Server-side Reasoning:** Implementing in-context learning and generating region-specific alerts requires retrieving information from multiple sources. When new, relevant data is received–whether from a disaster manager or a user–the system must process and classify it according to the message category. To do this effectively, a robust reasoning module is required to analyze the incoming information, assess the nature of the message, and determine its severity. This analysis is essential for generating accurate alerts and warnings. The system must identify the affected regions and relevant user categories by analyzing its user base and geographic data. It should then disseminate region-specific alerts that include customized, actionable instructions along with a severity score. The generated messages must integrate both the local context of individual users and the broader regional context to ensure relevance and clarity.
- **Memory Capability and Message Comprehension:** The system uses external tools and APIs to retrieve relevant information and evaluates the retrieved data to determine whether additional data should be fetched from websites and social media platforms. The database functions as the system's long-term memory, while the collected data-along with the current states of fishing vessels at sea serves as contextual information forming the system's short-term memory. This short-term memory also includes recently fetched information from various sources. Any newly acquired data deemed useful for future scenarios is stored in the database for ongoing processing and analysis, enabling the system to generate more accurate and actionable instructions during future events.

  Context Engineering: Context engineering enables the system to perform context reasoning, which is central to delivering contextually relevant and customized information. To achieve this, information from the client side must also be incorporated to better understand the local context. An essential component in this process is *context embedding*, which helps the system interpret both the meaning and context of incoming data. For example, a user's current location can be combined with retrieved warning information to produce personalized alerts. In this way, meaning is derived and decisions are made through the analysis of information aggregated from multiple sources.
- **Client Reasoning and Speech Synthesis:** Voice-enabled alert systems are significantly more effective than text messages, as users may not consistently check their mobile devices or read messages promptly. If the system can deliver audible alerts, it is more likely to capture the attention of the target community. Therefore, multilingual text-to-speech functionality is essential for delivering effective alerts during emergency situations. To support alert dissemination and determine whether a message should be played, a client-side reasoning module is also required. This module enables the system to decide in which languages the content should be presented, ensuring accessibility and relevance for diverse user groups.
- **Stakeholders and Users:** The user community includes coastal communities, fishing vessel owners, fishermen family, disaster mangers, and fishermen who venture into the ocean for fishing.

Apart from the above, the system utilizes the Model Context Protocol (MCP) to standardize access to external tools for retrieving relevant information from sources outside the internal memory. MCP adopts a client-server architecture and is composed of an MCP host, MCP client, and MCP server. These components collaborate with external tools and data sources to ensure the correct and designated outputs from the system. Due to its capacity to standardize interactions between AI models and external tools–including data retrieval and operation execution–MCP helps overcome the limitations of manual API integration and plugin-based interfaces, enabling more autonomous and intelligent agent workflows [22]. Hence, in our prototype implementation, we adopted this approach to provide access to external tools.

Fig. 2 illustrates how MCP can be utilized in the intelligent

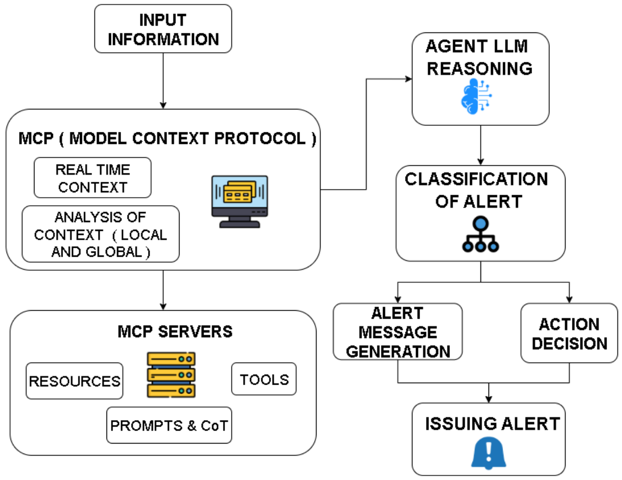

Fig. 2. MCP Enabled Alert System

automated alert dissemination system. MCP supports standardized integration of external tools and resources and even offers the flexibility to use multiple LLMs. These tools allow the MCP server to invoke external services and APIs, interact with other systems, and return the required information. For example, if the system needs to collect information from websites (e.g., INCOIS or IMD), MCP facilitates the use of the appropriate API, retrieves the data, and delivers it to the host for processing. MCP servers and tools enable the model to autonomously select and use external functionalities based on context, as these tools follow a standardized, modular, reusable, and easily accessible structure [22]. Overall, MCP provides the flexibility needed to streamline the processes required for the LLM agent to perform its designated tasks.

V. Implementation

Fishermen and coastal communities must receive customized information to enable them to take appropriate action upon receiving a warning. In addition to prioritizing the message, it is essential to assign a severity score during dissemination. Furthermore, wave height and location-specific intelligence are critical when the message is intended for coastal fishing communities. An effective and intelligent alert system must support both location-based and service-based alerts. It should also be capable of integrating local context into its alerting logic. For example, if a fisherman initiates a help request, the system should be able to process the information locally at the edge, determine the appropriate priority level, and generate suitable messages. These messages should be sent to nearby fishing vessels and, if necessary, broadcast as situation-based emergency alerts to all relevant emergency contacts. The backend of the Fisherman Alert System collects, categorizes, and reformulates government-issued marine threat notices into clear voice alert messages. It operates on a Multi-Channel Processing agent architecture. Developed in Python, the service runs continuously, retrieves new alerts on a predefined schedule, processes them through language models, and delivers the output via a RESTful API.

A. Data Ingestion and Preprocessing

To send alerts to fishermen and coastal communities, it is essential to retrieve information from existing maritime information dissemination websites. For implementation, marine threat alerts are fetched from the Indian National Centre for Ocean Information Services (INCOIS), which provides real-time hazard updates on its live website. To automate data retrieval, the backend employs a headless browser powered by Selenium WebDriver. This approach is necessary because the hazard data table is embedded within an IFRAME and rendered using client-side JavaScript, making it inaccessible through static HTML scraping alone.

Once the page and all its resources are fully loaded, the system identifies and extracts the relevant HTML table containing structured information such as states, districts, types of alerts, messages, and issuing dates. Since the table layout includes merged cells, appropriate state tracking is applied during parsing to ensure that each row is correctly associated with its corresponding geographical metadata. The extracted data is then normalized, date values are converted to the ISO 8601 format, and the results are stored in a JSON file. This file serves as a local cache that the API uses to handle all alert queries. The data collection process is scheduled to run at regular intervals (every 30 minutes) using a job scheduler.

B. Model Context Protocol (MCP) Agent Processing

The core of the alert processing system is a multi-channel, agent-based architecture known as MCP. Built using the Pydantic-AI toolkit, this layer manages the full lifecycle of small language model agents that transform raw alert text. Each agent operates based on a structured prompt that guides it through a step-by-step process: reading the technical marine notice, rewriting it into a plain, human-readable message, evaluating the seriousness of the alert, and adding useful tags such as safety tips and emergency contact numbers.

To ensure consistent responses and robustness against errors, each agent runs within an asyncio context managed by the MCP servers. When the server starts, a coroutine is launched to keep the server context alive indefinitely, allowing the language model to remain hot-loaded and responsive. As a result, inference-time delays are minimized, and the system can seamlessly scale by adding new agents as needed.

The use of the MCP server provides many advantages: it abstracts the complexity of the Asynchronous I/O and the model state, allowing the real-time message to pass between the components, and the main app supports concrete user sessions without blocking the thread. This is particularly important for public safety systems that must be on a scale to handle requests together during high-alert scenarios, such as cyclonic storms or tsunami warnings.

C. Alert Filtering and LLM Processing

Once the data is collected and stored locally, the backend system applies relevant filtering and language model-based

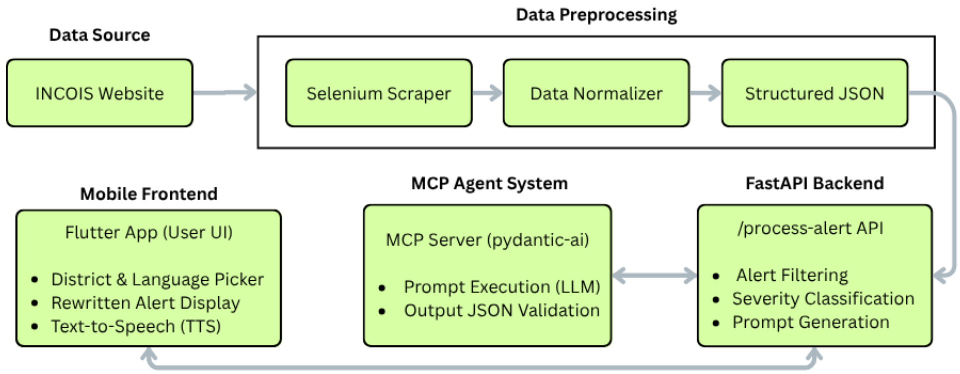

Fig. 3. System Implementation Workflow

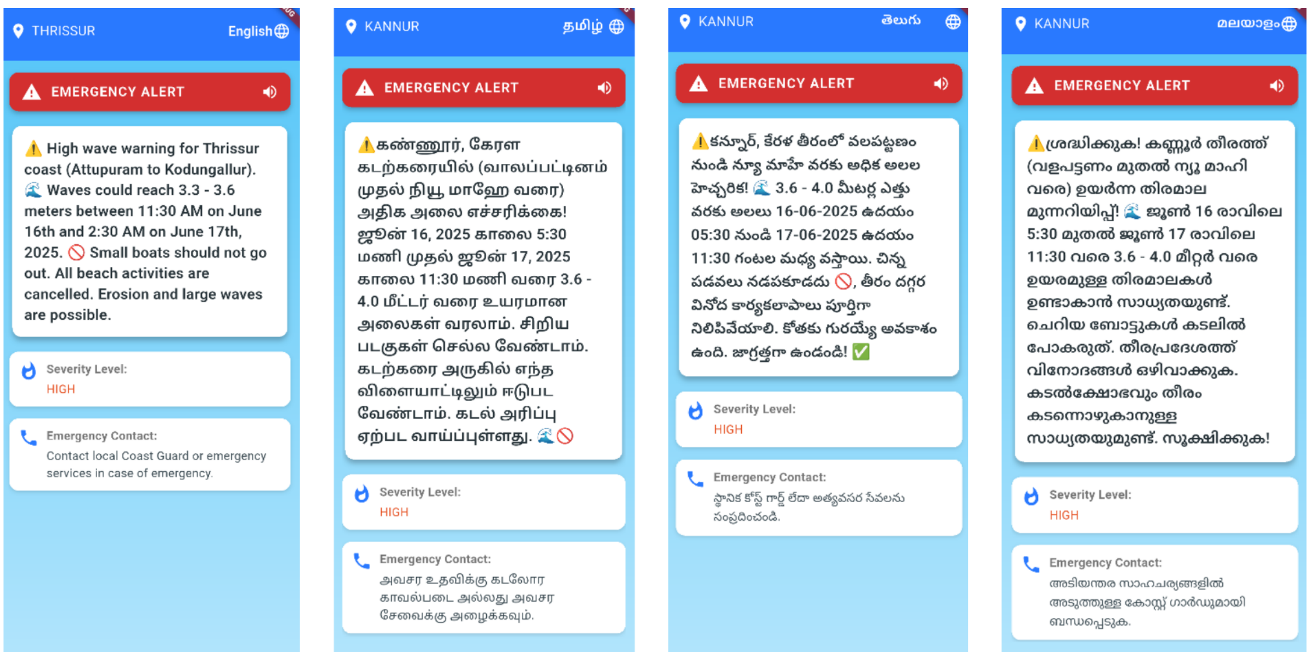

Fig. 4. Voice Alert in Multiple Languages

interpretation before generating the final output. This processing pipeline transforms raw institutional alerts into meaningful warnings tailored to individual, localized, and targeted audiences–specifically fishermen and coastal inhabitants, who often lack access to technical terminology or multilingual safety resources.

The filtering process begins with a user query issued to the /process-alert endpoint, which accepts criteria such as the target district. Additional alternative parameters include: the desired language (language_code) of the output. Internally, the system locally loads a cache JSON dataset that consists of all scraped alerts. Each alert entry is subject to a multi-step assessment. First, a case-insensitive match is done against the name of the district. Next, the alert message is classified by severity using the rule-based classification mechanism. A predetermined keyword-to-wide mapping translates into known danger details (eg, ”Cyclone,” high wave, ”strong wind”) into one of the four severity levels: low, medium, high, or critical. This allows the generalized classification system to apply threshold-based filtering, even if raw data can use different or inconsistent vocabulary. Once an alert is fetched, the system gives a structured prompt to the language model agent. This signal includes relevant information such as geographical location, alert classification, date of threat, and raw messages. This prompt is then passed to the MCP agent, which internally invokes a preconfigured Gemini large language model. The agent is assigned to make a determined multi-step change, as defined in its system prompt: understanding the alert, rewriting it in a simple natural language, classifying its severity, and generating safety advice and emergency contact recommendations.

The MCP agent runs in a continuous asynchronous context to reduce the inference delay and support concurrent processing. Once the model responds, the output is parsed using

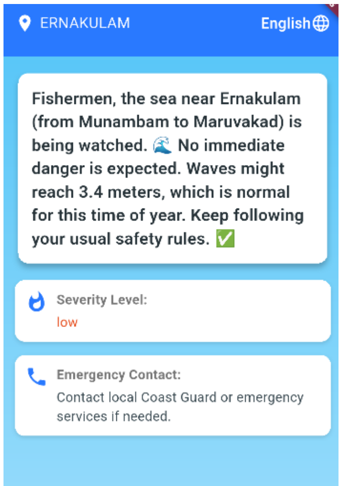

Fig. 5. No voice alert UI

regular expressions to extract the JSON structure from the text-generated. The schema includes mandatory fields such as rewritten_massage, is_alert, and severity_level, and alternative fields such as safety_guidence and emergency_contact_info. The use of a language model within the MCP agent framework offers several advantages over traditional rule-based or template-based rewriting systems. First, it enables reference-sensitive adaptations that take into account the specific danger type, location, and target audience. Second, it supports multilingual transformation, allowing alerts to be delivered in local languages such as Tamil, Malayalam, and others. Third, the model is capable of synthesizing actionable safety guidance even when the source message lacks clear instructions, thereby increasing the practical utility and relevance of the alerts.

The final output of the alert processing pipeline is a structured response that includes the converted message, relevant metadata, and safety recommendations. This output is returned to the front end via a JSON API response, which can be rendered in a mobile interface in the user's preferred language. The workflow of the complete system is illustrated in Fig. 3.

D. Frontend Implementation: Mobile Interface for Multilingual Alert Delivery

The frontend of the fishermen's alert system is implemented as a cross-platform mobile application using the Flutter framework. Upon startup, the application sends a request to the backend API, querying the latest available alerts for the user's district. This request includes both the geographical location (district name) and the user's preferred language, allowing the backend to return alerts that have already been processed by the MCP agent pipeline.

The application maintains a predefined list of coastal districts and supports five language options: English, Hindi, Tamil, Malayalam, and Telugu. Language selection is handled using standardized language codes, which map directly to the translation options supported by the backend agent. When a user selects a new district or switches languages, the frontend automatically triggers a new request for the backend, ensuring that the displayed alert is the latest and linguistic correct. The system also includes an integrated text-to-speech (TTS) engine, which reads alert messages if it is classified as high or severe. If the severity is low then it just displays the message and does not read loudly. This feature is important for users who may have limited literacy or who are occupied with hands-on activities such as fishing.

E. Results

Fig. 4 depicts the output of the Fisherman Alert System for various coastal districts. High-severity alerts, such as those for Thrissur and Kannur, are shown with clearly rewritten messages in multiple regional languages, along with safety guidance and emergency contact information. These alerts are also read aloud using the built-in text-to-speech (TTS) engine due to their elevated risk classification. In contrast, the Ernakulam district received a low-seriousness warning, where the system displays the message without triggering audio output. This conditional voice generation ensures that only actionable and important messages are amplified, avoiding unnecessary alerts. As depicted in Fig. 5, system intelligently disseminated low alerts.

VI. Limitation and Future Work

There are several limitations to the current study. First, the authors assume reliable connectivity with end users, including those located at sea. However, fishing vessels often experience limited or intermittent network access, which can affect the consistency and availability of local context data. A second limitation is the lack of consideration for computational efficiency on end-user devices. Most vessels have limited processing capabilities, making client-side reasoning and alert generation challenging. Therefore, further research is needed on the deployment of language models on edge devices that can operate effectively under constrained computing resources and intermittent connectivity.

In the current study, a multi-channel protocol is employed. However, to develop a full-fledged data dissemination framework, there is a need for multi-agent systems capable of handling specific, modular tasks. Designing collaboration protocols for communication and coordination among these agents is essential for generating and disseminating disaster alerts that are both context-aware and situation-aware. This also involves the activation of opportunistic communication mechanisms in intermittently connected environments, particularly relevant when fishermen rely on the OceanNet network for deep-water communication [7]. Hence, future work will focus on using Small Language Models (SLMs), which are designed to operate efficiently on devices with limited computational resources. SLMs are particularly useful in environments such as the ocean, where real-time processing is critical. This makes them well-suited for deployment on mobile devices, Internet of Things (IoT) systems, and other edge computing platforms [23]. SLMs aim to achieve high performance while minimizing computational and memory requirements.

Similarly, since the system must operate in a dynamic environment such as the ocean where link breakages are common, there is a need to move towards intelligent collaborative agents capable of performing real-world actions. This transition requires a shift from LLMs to Large Action Models (LAMs), which are specifically designed for action generation and execution in dynamic environments [24]. This will enhance the system's autonomy and reduce the need for human intervention in automated alert dissemination, making it more responsive to dynamic changes–especially during emergency situations where every minute counts. Additionally, as part of future work, it is crucial to validate the generated alerts with actual community members and disaster managers before deploying the system for real-world testing.

VII. Conclusion

Enhancing the safety of coastal fishing communities is essential for their overall growth and empowerment. Timely emergency alerts and actionable information are therefore crucial in advance of any forecasted emergency situation. This paper presented a novel LLM-enabled AI agent system capable of collecting information from diverse sources-including humans, websites, and social media to provide intelligent, targeted alerts to coastal communities, with messages customized according to severity. The paper described the architecture of the proposed LLM- and MCP-enabled intelligent autonomous alert dissemination system and outlined the current limitations of the study along with directions for future research. Initial results show that the proposed solution is a promising approach for alerting fishermen in their local languages, enabling them to understand the message and its urgency. Such targeted voice alerts in local languages help them take appropriate action to enhance their resilience.

Acknowledgment

We sincerely express our deep gratitude to our beloved Chancellor Dr. Sri. Mata Amritanandamayi Devi (AMMA), for inspiring us to undertake this work.

References

[1] V. Mody, V. Mody, and S. Parekh, "Distress–an application for emergency response and disaster management," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*. IEEE, 2020, pp. 830–836.

[2] R. Chen, J. Coles, J. Lee, and H. R. Rao, "Emergency communication and system design: The case of indian ocean tsunami," in *2009 International Conference on Information and Communication Technologies and Development (ICTD)*. IEEE, 2009, pp. 300–309.

[3] J. S. Jeyanathan, K. N. Kumar, P. V. Reddy, C. V. Chaithanya, M. N. Kumar, and P. Manikandan, "Gps assistive communication tool for fishermen community using internet of things," in *2023 8th International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 2023, pp. 361–364.

[4] S. Anand, M. Enayati, D. Raj, A. Montresor, and M. V. Ramesh, "Internet over the ocean: A smart iot-enabled digital ecosystem for empowering coastal fisher communities," *Technology in Society*, vol. 79, p. 102686, 2024.

[5] M. Angove, D. Arcas, R. Bailey, P. Carrasco, D. Coetzee, B. Fry, K. Gledhill, S. Harada, C. von Hillebrandt-Andrade, L. Kong *et al.*, "Ocean observations required to minimize uncertainty in global tsunami forecasts, warnings, and emergency response," *Frontiers in Marine Science*, vol. 6, p. 350, 2019.

[6] S. Anand, D. Raj, A. A. Sai, S. N Rao, and M. Vinodini Ramesh, "Techno-social synergy for disaster resilience in coastal communities: A sustainable approach," in *Proceedings of the 24th International Conference on Distributed Computing and Networking*, 2023, pp. 366–371.

[7] S. N. Rao, M. V. Ramesh, and V. Rangan, "Mobile infrastructure for coastal region offshore communications and networks," in *2016 IEEE Global Humanitarian Technology Conference (GHTC)*. IEEE, 2016, pp. 99–104.

[8] Y. Cheng, C. Zhang, Z. Zhang, X. Meng, S. Hong, W. Li, Z. Wang, Z. Wang, F. Yin, J. Zhao *et al.*, "Exploring large language model based intelligent agents: Definitions, methods, and prospects," *arXiv preprint arXiv:2401.03428*, 2024.

[9] M. Wooldridge, *An introduction to multiagent systems*. John wiley & sons, 2009.

[10] V. Sujitha, V. Anandhan, S. Anukeerthana, A. Dhanusriya, and M. Manibharathi, "Border security system for fishermen using radio frequency technique with live streaming," in *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*. IEEE, 2023, pp. 1246–1250.

[11] A. P. Ramesh, R. N. Paranthaman, M. Porkodi, V. Aruna, and S. Rohini, "Iot assisted fisherman aid to detect borders and alert system by using intelligent gps technology," in *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*. IEEE, 2023, pp. 1–6.

[12] T. Yano, M. Ando, K. Otsu, and T. Izumi, "Verification of the effects of voice and personalized disaster information from an agent on awareness that disaster is relevant to oneself," in *International Conference on Human-Computer Interaction*. Springer, 2023, pp. 548–557.

[13] H. Bean, B. F. Liu, S. Madden, J. Sutton, M. M. Wood, and D. S. Mileti, "Disaster warnings in your pocket: How audiences interpret mobile alerts for an unfamiliar hazard," *Journal of Contingencies and Crisis Management*, vol. 24, no. 3, pp. 136–147, 2016.

[14] M. V. G. Kumpatla, G. Keerthi Tej, G. Sahil, and S. Anand, "Enhanced road safety through ai-enabled voice assistance," in *Fifth International Conference on Computing and Network Communications*. Singapore: Springer Nature Singapore, 2025, pp. 147–160.

[15] P. Nedungadi, G. Veena, K.-Y. Tang, R. R. Menon, and R. Raman, "Ai techniques and applications for online social networks and media: Insights from bertopic modeling," *IEEE Access*, 2025.

[16] M. Venugopal, V. Madhavan, R. Prasad, and R. Raman, "Transformative ai in human resource management: enhancing workforce planning with topic modeling," *Cogent Business & Management*, vol. 11, no. 1, p. 2432550, 2024.

[17] R. Raman, A. Sreenivasan, M. Suresh, A. Gunasekaran, and P. Nedungadi, "Ai-driven education: a comparative study on chatgpt and bard in supply chain management contexts," *Cogent Business & Management*, vol. 11, no. 1, p. 2412742, 2024.

[18] X. Li, S. Wang, S. Zeng, Y. Wu, and Y. Yang, "A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges," *Vicinagearth*, vol. 1, no. 1, p. 9, 2024.

[19] D. Singh, D. Piplani, S. Shinde, and S. Karthik, "mkrishi fisheries: a case study on early warning system (ews) for disaster communication and management," in *2016 IEEE International Symposium on Technology and Society (ISTAS)*. IEEE, 2016, pp. 1–6.

[20] S. Anand, S. N. Rao, A. D. Nandan, B. M. Gopal, and M. A. Das, "Decision support system for spontaneous volunteers during emergency response," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2024, pp. 1–7.

[21] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen, "Synthetic prompting: Generating chain-of-thought demonstrations for large language models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 30 706–30 775.

[22] X. Hou, Y. Zhao, S. Wang, and H. Wang, "Model context protocol (mcp): Landscape, security threats, and future research directions," *arXiv preprint arXiv:2503.23278*, 2025.

[23] Q. Zhang, Z. Liu, and S. Pan, "The rise of small language models," *IEEE Intelligent Systems*, vol. 40, no. 1, pp. 30–37, 2025.

[24] L. Wang, F. Yang, C. Zhang, J. Lu, J. Qian, S. He, P. Zhao, B. Qiao, R. Huang, S. Qin *et al.*, "Large action models: From inception to implementation," *arXiv preprint arXiv:2412.10047*, 2024.