scientific reports

Check for updates

OPEN

Structured AI decision-making in disaster management

Julian Gerald Dcruz[1]✉, Argyrios Zolotas[1]✉, Niall Ross Greenwood[2] & Miguel Arana-Catania[1,3]

With artificial intelligence (AI) being applied to bring autonomy to decision-making in safety-critical domains such as the ones typified in the aerospace and emergency-response services, there has been a call to address the ethical implications of structuring those decisions, so they remain reliable and justifiable when human lives are at stake. This paper contributes to addressing the challenge of decision-making by proposing a structured decision-making framework as a foundational step towards responsible AI. The proposed structured decision-making framework is implemented in autonomous decision-making, specifically within disaster management. By introducing concepts of *Enabler* agents, *Levels* and *Scenarios*, the proposed framework's performance is evaluated against systems relying solely on judgement-based insights, as well as human operators who have disaster experience: victims, volunteers, and stakeholders. The results demonstrate that the structured decision-making framework achieves 60.94% greater stability in consistently accurate decisions across multiple *Scenarios*, compared to judgement-based systems. Moreover, the study shows that the proposed framework outperforms human operators with a 38.93% higher accuracy across various *Scenarios*. These findings demonstrate the promise of the structured decision-making framework for building more reliable autonomous AI applications in safety-critical contexts.

Keywords Responsible AI, Structured decision-making

In disaster management, decision-making can be categorized into three phases: the pre-disaster phase, which involves sending warnings and planning evacuation procedures; the disaster phase, focused on rescue and relief operations; and the post-disaster phase, centred on assessing damage for reconstruction or rehabilitation activities[1]. With the rapid advancement of artificial intelligence (AI), unmanned aerial vehicles (UAVs), and satellite imagery, significant efforts have been made to apply AI in providing judgment insights across each phase of the disaster management cycle[2]. While there have been initiatives to incorporate autonomous decision-making in tasks such as coordinating relief operations[3], these systems are scrutinized in safety-critical contexts due to the potential harm they may cause. This scrutiny has led to a paradox: *systems designed to enhance safety and efficiency are questioned for the risks they might introduce, particularly when errors or biases in autonomous decisions could have severe consequences*. This paradox underscores the tension between the advantages of automation and the need for meaningful human oversight to ensure accountability, fairness, and safety[4]. To address the challenges of reliable and justifiable decisions in designing autonomous decision-making systems for safety-critical domains like disaster management, it is crucial to incorporate structure into AI decision-making.

The work in this paper stems from a research project that proposed and designed a structured decision-making framework for autonomous decision-making in safety-critical domains. The work particularly contributes the following:

- Designing and developing a structured decision-making framework for autonomous decision-making within disaster management.
- Developing an autonomous decision-making agent that makes use of the previous framework to enhance its decisions in disaster management.
- Conducting a human evaluation study to effectively evaluate the autonomous decision-maker with human operators.

The project's codebase is publicly available in the project's repository (https://github.com/From-Governance-To-Autonomous-Robots/Autonomous-Governance-in-Disaster-Management).

[1]Faculty of Engineering and Applied Sciences, Cranfield University, Bedford, UK. [2]Ageno School of Business, Golden Gate University, San Francisco, USA. [3]Digital Scholarship at Oxford, University of Oxford, Oxford, UK. ✉email: juliangeralddcruz@outlook.com; a.zolotas@cranfield.ac.uk

The structure of the paper is as follows: "Related work" provides a brief survey of related work in the current literature. Section "Framework for structured decision making" describes the proposed framework, followed by a detailed methodology of the approach in "Methods". Section "Results" presents and analyzes the experimental results, including a comparative analysis between human evaluation and the RL decision maker. Finally, "Conclusions" draws conclusions from the study.

Related work

In disaster management, decision-making can be inherently complex, often unstructured, dynamic, and unpredictable. This process demands significant judgment and insight, frequently involving coordination between multiple local and government agencies, such as volunteer organizations and governmental authorities. Such unstructured decision-making can lead to unequal or delayed responses to victims' needs, as decisions are frequently based on immediate perceptions rather than structured data. For instance, the response to the Fukushima disaster highlighted the challenges faced by rescue teams in making critical decisions under uncertainty[5]. Similarly, during a major railway accident in the UK, poor coordination among agencies significantly hampered the disaster response efforts[6].

Over the years, the application of machine learning in disaster management has significantly improved decision-making processes, particularly by enabling faster response times. The advent of surveillance drones, satellite imagery, and various data modalities has inundated decision makers with extensive data, necessitating the use of machine learning algorithms to generate explainable insights.

Previous research has explored various machine learning techniques to provide explainable insights during the disaster response phase, thereby supporting emergency response activities[7–10]. In this context,[11] proposed a framework utilizing satellite imagery from six types of natural disasters to identify the primary causes of damage in affected areas, achieving 99.59% accuracy and facilitating effective interventions. Similarly,[12] introduced a multimodal deep learning model that integrates text and image data from social media, outperforming single-modality models. These studies demonstrated the viability of leveraging social media posts to triage victim requests and support decision-making in emergency response. Additionally, machine learning techniques have proven valuable in aiding decisions related to reconstruction and rehabilitation during the post-disaster phase.[13] trained a CNN model combined with an ensemble max-voting classifier to assess flood damage in Texas using aerial images, achieving 85.6% accuracy and an F1-score of 89.06%.[14] developed a CNN model to assess the impact of water-related disasters by segmenting topographical features in pre- and post-disaster satellite images, identifying regions with significant changes. These studies demonstrated that machine learning techniques can expedite the analysis of satellite and drone data, reducing reliance on expert opinion and mitigating delays or inaccuracies in post-disaster decision-making. Inspired by these works, this paper utilizes the CrisisMMD Multimodal Twitter dataset[15], an image-text dataset collected during various natural disasters, to address decision-making in the disaster phase of the disaster management cycle. CrisisMMD is the de-facto benchmark for image–text fusion during fast-moving natural hazards and contains 16 058 tweets paired with 18 082 images spanning seven 2017 disasters. Although modest in size, CrisisMMD remains the only openly licensed dataset in which every tweet–image pair is simultaneously annotated for (i) informativeness, (ii) humanitarian category, and (iii) damage severity. That unique tri-label design explains why virtually every multimodal disaster-response study since 2018 tests on it. The current state-of-the-art–transformer models with cross-modal attention and self-attention (e.g.,[16])–pushes the macro-F_1 ceiling to roughly 92 %. Even so, performance is capped by two structural flaws: (1) noisy crowdsourced labels, and (2) temporal homogeneity–every event in the corpus happened in 2017. Minority classes such as missing people and vehicle damage remain chronically under-represented; most authors collapse them into broader buckets, a stop-gap that is anything but ideal for real-world triage. Additionally, the xBD dataset[17], containing pre- and post-event satellite imagery for various disasters, and the RescueNet dataset[18], comprising post-disaster UAV images from multiple impacted regions, are utilized to address decision-making in the post-disaster phase. The xBD dataset is the largest fully-annotated satellite dataset for post-disaster building assessment: 22 068 image pairs, 850 736 building polygons, four-level "Joint Damage Scale", and 80/10/10 train–test–holdout splits over 19 events and 45k km^2 of ground coverage. It underpins the xView2 Challenge and therefore anchors most modern work on large-scale change detection. Gupta et al.'s baseline CNN hits 0.59 macro F1; hierarchical transformer architectures[19] now exceed 0.71 by explicitly modelling multi-resolution context and temporal deltas. Yet xBD is still EO-only, dominated by U.S. and Western Pacific imagery, and heavily skewed toward the "no-damage" class (8× the next largest), so models trained here rarely transfer cleanly to SAR imagery or under-represented geographies. The RescueNet dataset fills the altitude gap with 4 494 4096 × 3072 UAV frames captured after Hurricane Michael, labelled at pixel-level for 11 semantic classes (from "road-blocked" to four granular building-damage states) and split 80/10/10 for train/val/test. Baselines on DeepLabv3 + ResNet-101 reach 81% mean IoU[18], but performance collapses on the minority "road-blocked" and "pool" classes–evidence that high-resolution alone does not solve class imbalance.

However, these studies have not succeeded in addressing the challenges in the operations of disaster management which involves different agencies, including local governments, non-governmental organizations, and international bodies, and these agencies often follow varied protocols and communication standards, which leads to inter-agency coordination challenges. This lack of coherence can result in duplicated efforts, resource misallocation, and delays in critical response activities. During Hurricane Katrina, for example, misalignment between federal, state, and local authorities led to substantial delays in providing aid and evacuating affected individuals[20]. Furthermore, current disaster management practices often struggle with information overload and the integration of diverse data sources. Stakeholders must frequently make rapid decisions based on incomplete or rapidly changing information, which can lead to potential oversights and mistakes[21]. Additionally, prolonged exposure to disaster-related decision-making can result in decreased accuracy and effectiveness among stakeholders, a phenomenon referred to as community disaster fatigue, particularly manifesting as a

sense of defeatism[22]. In response to these challenges, recent years have seen significant advancements in the introduction of AI for autonomous decision-making in disaster management.[3] explored the coordination of semi-autonomous robot teams for disaster relief, focusing on the Situated Decision Process for managing tasks under uncertainty. They identified critical challenges in integrating reactive and deliberative planning, ensuring that autonomous vehicles execute their missions as intended without direct human oversight. Furthermore,[23] examined the integration of big data analytics with IoT in disaster management, highlighting the difficulties in managing large volumes of data, making timely decisions, and maintaining the reliability of autonomous systems under crisis conditions.

Most of these works focus on replacing the human in the loop for safety-critical decisions in disaster management.[24] underscored the ethical implications of AI in safety-critical environments, particularly in disaster response contexts, and discussed the necessity of governance frameworks to ensure responsible, transparent, and fair decision-making processes. Since machine learning (ML) decisions are heavily data-driven and inherently unpredictable, these systems may exhibit significantly different behaviours in response to nearly identical inputs, making it difficult to define 'correct' behaviours and ensure safety in advance[25]. Moreover, scholars have identified potential safety risks arising from the interaction between AI systems and their users, particularly due to automation bias. This bias occurs when humans attribute greater credibility to automated decisions because of their perceived objectivity, leading to complacency and less cautious behaviour when interacting with AI systems[26,27]. Autonomous decision-making significantly reduces human control, particularly in safety-critical decisions where AI systems are responsible for actions that impact human lives. Current legal frameworks typically place the responsibility for decision-making on humans, treating machine learning operations as tools that assist in the decision-making process[28,29]. However, since AI largely depends on processes that independently develop and modify their own rules, human oversight is diminished. This reduction in control makes it unreasonable to hold human stakeholders or manufacturers fully accountable for the AI's actions, especially given the unpredictable nature of ML decisions, which implies that erroneous decisions made by AI are beyond the control and anticipation of both parties[30,31].

The deployment of robots for personal care services, for instance, has sparked significant worries about the potential compromise of patient autonomy and dignity, particularly when robots impose strict limitations on patient mobility to prevent dangerous situations, see[32,33]. Similarly, the introduction of autonomous weapon systems, including drones and unmanned aerial vehicles, aimed at enhancing the precision and effectiveness of military operations, has raised concerns about the ethical and legal implications of such technologies[34,35]. The delegation of authority to machines in making decisions about the use of lethal force, coupled with the lack of control over these autonomous systems, has also prompted significant ethical debates[36–38].

To address these ethical implications, scholars have been proposing governance frameworks for responsible AI[39–41].[42] proposed a governance framework for AI in oncology, where the decision-making on safety-critical decisions directly relates to the decision-making context of disaster management. The framework proposed by[42] emphasized ethical soundness, legal compliance, equity, and patient-centred care, with an emphasis on quality assurance, transparency, bias mitigation, stakeholder engagement, and continuous adaptation through regular audits and feedback loops.

Framework for structured decision making

This section presents the proposed framework for structured decision-making. The framework aims to address inter-agency coordination and data overloading challenges, and mitigate the defeatism often experienced by stakeholders in disaster scenarios. The framework specifically focuses on implementing a structured decision-making approach, by clearly defining roles for the agents (`Enabler` and `Decision Maker` agents) and how they interact to make decisions at each `Level` of the disaster management `Scenario`. The framework aims to optimize decision-making flow through traceability of decisions in emergency response, rehabilitation, and reconstruction activities.

From a high-level perspective, the proposed framework for structured decision-making in disaster management comprises `Enabler` agents and `Decision Maker` agents, which collaborate to effectively manage disaster-phase and post-disaster phase scenarios. A representation of this framework is shown in Fig. 1, which also compares the decision flow with that of a human decision-maker in disaster management. To implement structured decision-making, the decision flow is organized into distinct `Levels`, as illustrated in Fig. 2. This structured approach is visually represented as a tree-like structure, referred to as a `Scenario`. A `Scenario` consists of five nodes, depicted as grey-filled boxes in the figure. Each node, known as a `Level` within the framework, represents a stage where a critical decision must be made, potentially leading to a consequence or reward. The yellow-filled ellipses indicate the decision options available to the `Decision Maker` agent based on the data received at that `Level`. At each `Level`, the data is first processed by an `Enabler` agent before the `Decision Maker` can make a decision. The `Enabler` agent is an AI model trained to evaluate disaster-related data at that `Level`, providing judgment insights (an array of confidence scores for each decision option) to assist the `Decision Maker` in making informed decisions within a `Scenario`. The `Decision Maker` agent can be either a reinforcement learning (RL) algorithm or a human operator. However, if the `Decision Maker` is a human operator, the `Enabler` agent is not involved, as the human relies on their expertise and training to make decisions at each `Level`. Additionally, the blue-filled ellipses in the diagram represent the `Gather Additional Data` option, which serves as a real-world interaction point for the RL agent or human operator. This option allows for the acquisition of additional data at a given `Level`.

One `Scenario` comprises of the following levels:

- `Level`-1, where the data received at this stage is either informative or not informative to the disaster.
- `Level`-2, where the data received at this stage is related to a type of humanitarian effort.

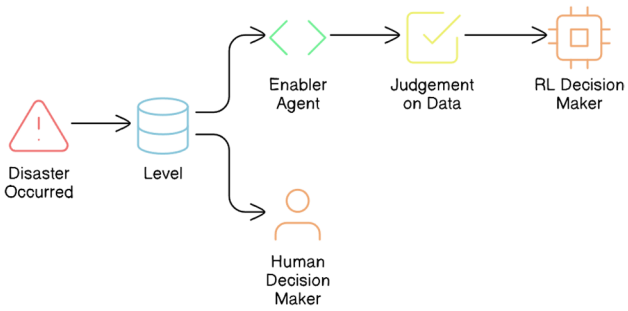


Figure 1. High-level overview of the proposed framework for structured decision-making in an autonomous system, compared with the decision flow of a human decision-maker.

- `Level-3`, where the data received at this stage is related to damage severity and was collected by victims or volunteers.
- `Level-4`, where the data received at this stage is related to damage severity and was collected by satellite imagery.
- `Level-5`, where the data received at this stage is related to damage severity and was collected by Unmanned Aerial Vehicles (UAVs).

Methods

This section details the methodology followed in the implementation and evaluation of the proposed framework. The framework involves `Enabler` agents and `Decision Maker` agents collaborating across multiple `Levels` in a tree-like `Scenario` structure. Each `Level` represents a critical decision point, where `Enabler` agents process disaster-related data to assist the `Decision Maker`, which can be either a reinforcement learning (RL) algorithm or a human operator. This section details each `Level`, the training methodology of the RL agent, the evaluation metrics used, and the logic behind the web application designed for the evaluation, as well as the data used.

Dataset overview

Our implementation focuses on decision-making during the disaster phase and post-disaster phase in disaster management. For disaster-phase decisions related to emergency response activities such as search, rescue, and relief operations, the CrisisMMD dataset[15] (a multimodal image-text dataset on disaster response activities collected from Twitter) is employed. CrisisMMD contains 16 058 tweets paired with 18 082 images spanning seven 2017 disasters. Every tweet–image pair is simultaneously annotated for (i) informativeness, (ii) humanitarian category, and (iii) damage severity. For post-disaster phase decisions involving damage assessment for reconstruction or rehabilitation activities, the xBD dataset (a dataset of images collected from satellites on pre- and post-disaster damage)[17] and the RescueNet dataset (a dataset of images collected from drones on pre- and post- disaster damage)[18] are used. The xBD dataset contains 22 068 image pairs, 850 736 building polygons, four-level "Joint Damage Scale", and 80/10/10 train–test–holdout splits over 19 events and 45k km^2 of ground coverage. xBD is still EO-only, dominated by U.S. and Western Pacific imagery, and heavily skewed toward the "no-damage" class (8× the next largest). The RescueNet dataset fills the altitude gap with 4 494 4096 × 3072 UAV frames captured after Hurricane Michael, labelled at pixel-level for 11 semantic classes (from "road-blocked" to four granular building-damage states) and split 80/10/10 for train/val/test.

These datasets are used to train the `Enabler` agents (which is discussed in more detail in the next section) that make up the structured decision-making. Each dataset will be presented in detail in the following sections together with each of the machine learning models using it.

Enabler agent

The `Enabler` agent supports the RL `Decision Maker` agent by providing judgment insights on the data received at each `Level` of a `Scenario`, enabling informed decision-making. This section outlines the methodology used to train the `Enabler` agents for the `Levels` corresponding to the disaster-phase and post-disaster phase of the disaster management cycle.

Disaster phase

From the proposed framework in Fig. 2, the decisions from `Level-1`, `Level-2`, and `Level-3` are for activities in the disaster phase related to emergency response services.

The CrisisMMD[15] dataset comprises image+text pairs with their corresponding annotations, as seen in Fig. 3. This dataset is utilized to train the three `Enabler` agents for the three `Levels`. The `Enabler` agent here is a multimodal multiclass classification model.

- For `Level-1`, the `Enabler` agent is trained to classify the image-text pair as either 'informative' or 'not-informative' regarding the disaster.
- For `Level-2`, the `Enabler` agent identifies the type of humanitarian aid represented by the image-text pair. The target classes are 'affected individuals,' 'infrastructure/utility damage,' 'other relevant information,' and

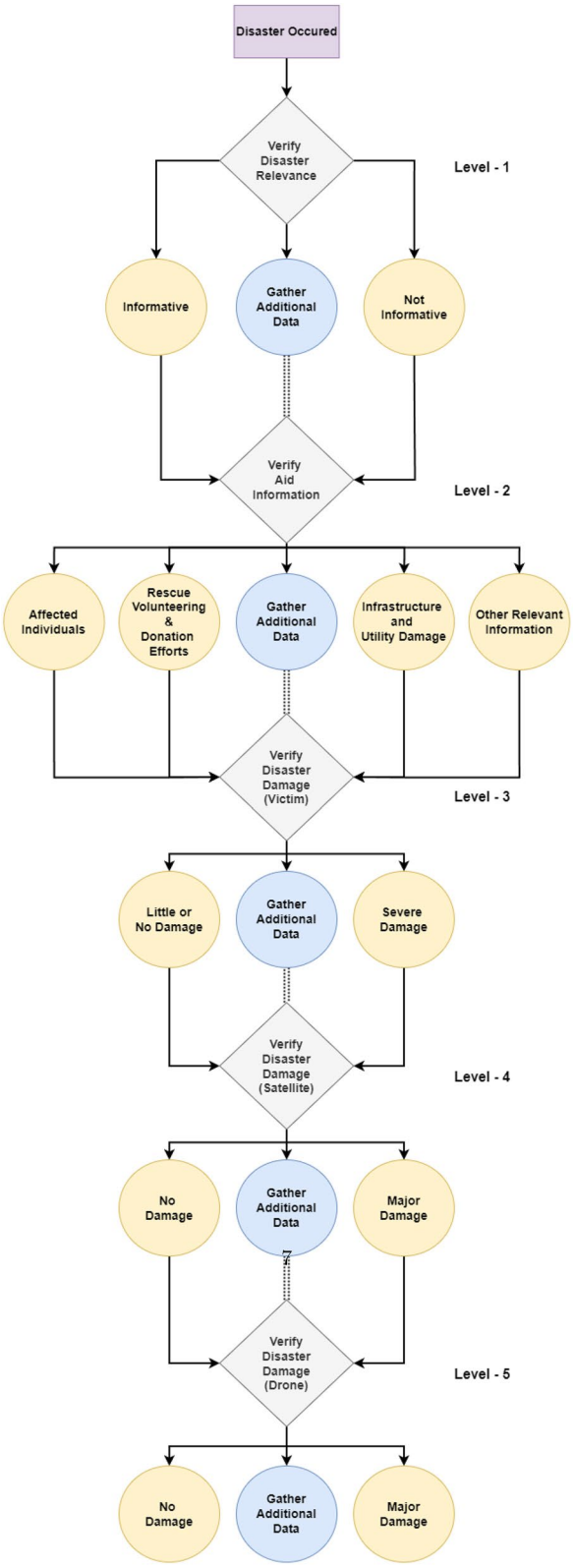


Figure 2. System design of the proposed framework for structured decision-making in disaster management.

(a) The #flood extent in Kalutara District, #Sri-Lanka, was captured by #TerraSARX on 30 May

(b) She got her stage name from a liquor store on a bottle of Harvey's Bristol Cream Sherry

(c) #foxnews #cnn #ap #Iran National Television (INTV) donates to the victims of the earthquake

(d) J.J. Barea teamed up with Mark Cuban to bring aid to Puerto Rico

(e) 5AM Thursday 9/7 Hurricane Irma update: The threat for major impacts for South Florida continues to increase. #FLwx

(f) NASA Tech Is Helping Find Mexico Earthquake Victims Buried in Rubble

Figure 3. Example data from the CrisisMMD dataset, used to train Enabler agents across Scenario Levels 1, 2, and 3 corresponding to the following data categories: (**a**) Informative, (**b**) Not-informative, (**c**) Affected individuals, (**d**) Infrastructure/utility damage, (**e**) Other relevant information, and (**f**) Rescue/volunteering efforts.

'rescue/volunteering efforts.' The labels 'injured or dead people' and 'missing or found people' were relabeled to 'affected individuals,' and 'vehicle damage' was relabeled as 'infrastructure/utility damage.' The 'not humanitarian' label was removed due to its irrelevance in the decision process.

- For `Level-3`, the `Enabler` agent assesses the severity of damage in the image-text pair. The target classes are 'little or no damage', and 'severe damage'. The label 'mild damage' was relabeled to 'little or no damage' to better reflect the context.

Inspired by[43], a multimodal model architecture consisting of a BiLSTM for text processing and a ResNet50 for image processing is designed, as seen in Fig. 4. The BiLSTM produces two sets of hidden states, one from processing the text forwards $\overrightarrow{h_t}$ and another from processing it backwards $\overleftarrow{h_t}$. These hidden states are concatenated to form the final hidden state for each time step $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$. The sequence of hidden states is then summarized using average pooling,

$$\text{avg\_pool} = \frac{1}{T}\sum_{t=1}^{T} h_t,$$

and max pooling,

$$\text{max\_pool} = \max_{t=1}^{T} h_t,$$

with the resulting vectors concatenated to form the text feature vector. Example: suppose $T{=}3$ and each $h_t \in \mathbb{R}^3$ is $h_1 = [0.2, 0.7, -0.1]$, $h_2 = [0.4, 0.1, 0.5]$, $h_3 = [-0.3, 0.6, 0.0]$. Element-wise averaging gives $\text{avg\_pool} = \frac{h_1+h_2+h_3}{3} = [(0.2+0.4-0.3)/3, (0.7+0.1+0.6)/3, (-0.1+0.5+0.0)/3] = [0.1, 0.467, 0.133]$. Element-wise max pooling yields $\text{max\_pool} = [0.4, 0.7, 0.5]$. Concatenating them produces a $2 \times 3 = 6$-dimensional text vector $[0.1, 0.467, 0.133, 0.4, 0.7, 0.5]$. This vector is then concatenated with the 256-dimensional image feature vector obtained from the ResNet50 model[44], forming a combined feature vector. The combined

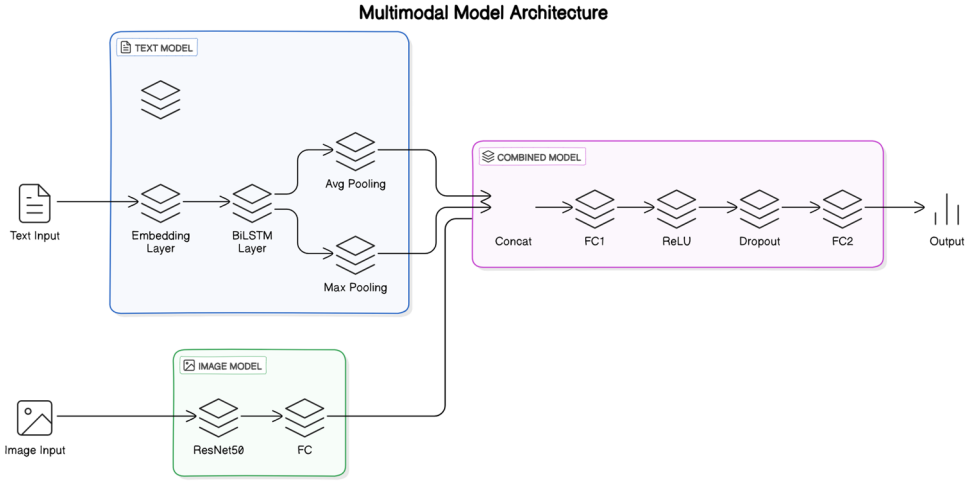


Figure 4. Multimodal model architecture.

Label	Total data count
Level-1	
Informative	9162
Not informative	8817
Level-2	–
Infrastructure and utility damage	3830
Rescue volunteering or donation effort	2231
Other relevant information	2528
Affected individuals	686
Level-3	
Severe damage	2212
Little or no damage	1314

Table 1. Class distribution for disaster phase labels in Level-1, Level-2, and Level-3.

vector, with a dimensionality of $2 \times \text{hidden\_size} + 256$, is passed through a fully connected layer with ReLU activation and dropout to reduce overfitting. Finally, the processed features are fed into another fully connected layer, producing the final output; the confidence scores for all classes are recorded as an array and stored as the judgement used later in a `Scenario`.

In preparing the dataset for training the `Enabler` agents, the Twitter text is cleaned to replace contractions with their full forms, standardize numbers, remove special characters, clean up social media-specific elements (like URLs, mentions, and hashtags), and remove common stop-words, this allows for a more uniform and simplified text.

A Keras Tokenizer[45] is fit on the cleaned Twitter text data, converting the text into sequences of integers that correspond to the words in the tokenizer's vocabulary. The tokenized sequences are then padded to have the same length which is necessary for consistent input to machine learning models. The Keras tokenizer uses a vocabulary size of 20,000 for `Level-1` and `Level-2`, and 10,000 for `Level-3`. In all cases, padded with a maximum sequence length of 20.

Additionally, pre-trained GloVe embeddings[46] are loaded that map each word in the tokenizer's vocabulary to its corresponding GloVe vector, enabling the model to leverage these rich word representations during training. The GloVe embeddings[47] help capture semantic relationships between words. They are pre-trained on a large corpus of text capturing a broad understanding of word relationships and meanings. They help in reducing the complexity of the data reducing the risk of overfitting, and they can help mitigate the problem of out-of-vocabulary (OOV) words.

The datasets for each `Level` in the disaster phase are stratified into a train set and validation set in an 80:20 ratio such that the proportion of each class in the target variable is preserved in both sets, as seen in Table 1.

The images are resized to 224x224, and random horizontal flip augmentation is applied using the TorchVision[48] library. Normalization is then performed to adjust the pixel values so that each colour channel has a mean of 0 and a standard deviation of 1.

The learning rates are 0.0001 and the number of epochs is 100 for the three `Levels`. The batch sizes are 4 for `Level-1`, and 32 for `Level-2` and `Level-3`.

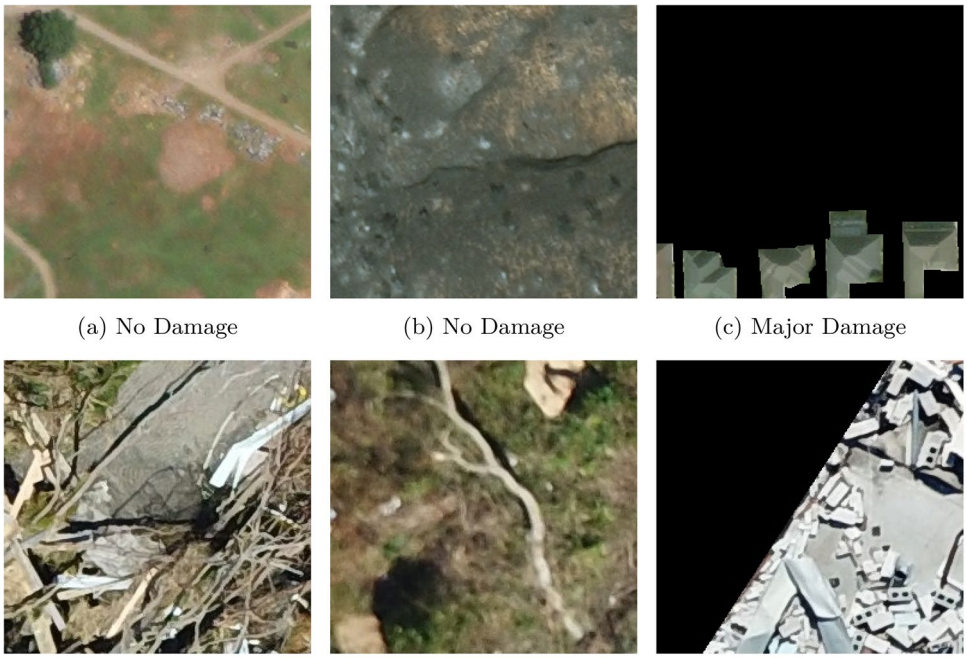
Post-disaster phase

The decisions of damage assessment made at `Level-4` and `Level-5` are crucial components of the post-disaster phase. In this phase, the `Enabler` is provided with images captured by satellites and drones. The `Enabler`, which is a multilabel image classification model designed to evaluate the severity of the damage depicted in the images. The xBD and RescueNet datasets, which are utilized for this task, provide segmentation annotations for each corresponding image.

- For `Level-4`, the `Enabler` agent is trained to identify 'no damage' and 'major damage' from the satellite images from the xBD dataset. The labels 'destroyed' and 'minor damage' were relabelled as 'major damage' to simplify training, and avoid confusion among labels of damage.
- For `Level-5`, the `Enabler` agent is trained to identify 'building destroyed' and 'building no damage' from the drone images from RescueNet dataset. The label 'building major damage' was relabelled to 'building destroyed', and the 'building minor damage' was relabelled to 'building no damage'. The following classes: 'background', 'water', 'vehicle', 'tree', 'pool', 'road clear', and 'road blocked' were ignored by replacing pixel values with 0 using their segmentation annotations, this was done to simplify the decisions to be made at this `Level`.

Although training a segmentation model would potentially yield higher accuracy, a multilabel classification approach is chosen, to simplify the training process. This is justified by the fact that an image can be associated with multiple labels. The model used for multilabel classification is built upon the ResNet-50 architecture, which uses deep convolutional layers and residual connections. Using a pre-trained ResNet-50 model as the base, takes advantage of the knowledge it has acquired from large datasets like ImageNet to extract meaningful features from the input images. To accommodate the multilabel classification task, the original fully connected layer of the ResNet-50 was replaced with a new fully connected layer that outputs a number of features corresponding to the total number of classes in the dataset. This modification allows the model to predict multiple labels for a single image. Additionally, a sigmoid activation function was applied to the output of this modified layer. The sigmoid function converts the raw outputs into probabilities, enabling the independent prediction of each class label, which is essential for multilabel classification tasks. This customized ResNet-50 model, integrated with a sigmoid activation function, serves as the `Enabler` agent and is trained on a well-structured dataset that accurately reflects the complex and multifaceted nature of post-disaster damage assessment.

Given that the dataset contains images with large resolutions, it is essential to ensure efficient training and faster learning. As part of this study, the images are divided into smaller patches of size 256×256 with a step size of 256, as seen in Fig. 5. To construct the dataset for training the multilabel classification model, the segmentation annotations were used to classify the images and generate a CSV file for both training and validation. The CSV file contains columns representing each class, as well as the image paths. Each image is recorded only once in the CSV file, and if an image is associated with multiple labels, the corresponding columns for those labels are populated with a value of 1, indicating the presence of those labels in the image. This method



(a) No Damage (b) No Damage (c) Major Damage

(d) Building No Damage (e) Building No Damage (f) Building Destroyed

Figure 5. Example data from the post-disaster phase, displaying the images with their corresponding annotation, used to train Enabler agents across Scenario Levels 3 and 4. (**a**–**c**) are satellite images from the xBD dataset, (**d**–**f**) are drone-captured images from the RescueNet dataset.

Label	Train data count	Validation data count
Level-4		
No damage	89,566	29,856
Major damage	38,601	12,355
Level-5		
Building no-damage	46,569	1974
Building destroyed	27,951	1232

Table 2. Class distribution for post-disaster phase labels in Level-4, and Level-5.

Component	Dimension
Expected length of prediction confidence array	4
Length of `Level` vector array	5
Total observation	**9**

Table 3. Observation space.

effectively structures the dataset for training the multilabel classification model. The train and validation dataset is constructed from the train and validation data provided from the original xBD dataset and RescueNet dataset, this is seen in Table 2.

The learning rates are 0.001 for `Level`-4, and 0.0001 for `Level`-5. The number of epochs is 100 and the batch sizes are 16 for the two `Levels`.

Enabler agent metrics

To evaluate the performance of the Enabler Agent model for the classification task, the following metrics were computed using the 'scikit-learn'[49] package in Python:

- **Precision**: The proportion of true positive predictions out of all positive predictions.
- **Recall**: The proportion of true positive predictions out of all actual positive instances in the dataset.
- **F1-Score**: The harmonic mean of precision and recall.
- **Accuracy**: The overall proportion of correct predictions out of all predictions made by the model.
- **Macro Avg**: The average of a metric (e.g., precision, recall, F1-Score) calculated independently for each class and then averaged.
- **Weighted Avg**: The average of a metric (e.g., precision, recall, F1-Score) calculated independently for each class, weighted by the number of true instances for each class.
- **Support**: The number of true instances for each class in the dataset.

The primary metric used to select the best model is the **Macro Average F1-score**, as it provides the model's performance across the classes by averaging the F1-scores for each class, and is well suited due to the imbalance in the dataset.

Reinforcement learning decision maker

The `Decision Maker` agent determines the activities to be undertaken at each `Level` in a `Scenario`. This section discusses the reinforcement learning `Decision Maker` agent, and the following section will discuss the human operator agent.

The reinforcement learning (RL) agent is what brings about autonomous decision-making in the framework. The judgements made by the `Enabler` agents at each `Level` inform the decision-making of the RL agent, which brings structure to the decision-making process, making it reliable and justifiable. This decision flow of the RL agent can be seen in Fig. 1.

The data for training and evaluation of the RL agent is prepared by running inference using the trained `Enabler` agents for each `Level` in a `Scenario` on their respective training and validation datasets, and then storing the predicted confidence scores for all classes along with the ground truth. These results are then utilized by the RL agent. The RL agent's environment is a custom Gymnasium[50], and is designed so that, when a decision is required at a `Level`, an unused random data record is fetched from the stored outputs for the `Enabler` agent at that `Level`. Since each `Level` has a varying number of possible actions that are expected, it is required that the RL agent environment has a different action space defined for each `Level`. One episode in the RL environment corresponds to a `Scenario`, with each step representing a `Level` within that `Scenario` based on the action taken.

The observation space for the RL agent is a one-dimensional array with a length of 9, a breakdown of the observation space is seen in Table 3. The first 4 elements consist of the prediction confidence array from the `Enabler` agent for that `Level`. Since the prediction confidence array can vary in length across different `Levels`, it is padded with zeros if its length is less than 4. The remaining 5 elements form the `Level` vector, which indicates the specific `Level` from which the observation is collected. For example, '10000' represents

Level-1, and '00001' represents Level-5. The final observation space is the concatenation of the padded prediction confidence array and the Level vector.

Each Level offers a varying number of possible actions, as shown in Table 4. Correct or incorrect actions can result in advancing to the next Level, with associated rewards or consequences. The RL environment is structured so that the action space is determined by the current Level. At the end of each step, if the current Level is completed, the action space for the next Level is set, ensuring that invalid actions from other Levels are not predicted by the RL agent. Additionally, each Level includes a Gather Additional Data action, which allows the Decision Maker agent to request new data for better decision-making while remaining at the current Level. This action does not advance the agent to the next Level but instead updates the observation space with new data, keeping the agent on the same Level for the next step.

- For Level-1, the action space includes: 0 which represents 'informative', 1 which represents 'not informative', and 2 which represents Gather Additional Data.
- For Level-2, the action space includes: 0 which represents'affected individuals', 1 which represents 'infrastructure and utility damage', 2 which represents 'other relevant information', 3 which represents 'rescue and volunteering efforts', and 4 which represents Gather Additional Data.
- For Level-3, the action space includes: 0 which represents 'little or no damage', 1 which represents 'severe damage', and 2 which represents Gather Additional Data.
- For Level-4, the action space includes: 0 which represents 'no damage', 1 which represents 'major damage', and 2 which represents Gather Additional Data.
- For Level-5, the action space includes: 0 which represents 'building no damage', 1 which represents 'building destroyed', and 2 which represents Gather Additional Data.

In a Scenario, each Level has an allocation of 5 credits to request additional data, requesting additional data can result in a minor penalty of−1. Correct decisions result in a reward of +1 and wrong decisions result in a penalty of−5.

An Advantage Actor–Critic (A2C) algorithm [51] with a multilayer-perceptron (MLP) policy was trained using the following hyper-parameters. The discount factor was fixed to $\gamma = 0.995$, emphasising delayed rewards. Each policy update processed $n_{\text{steps}} = 128$ environment interactions, a compromise between sample efficiency and return-variance reduction. The entropy coefficient was set to *ent_coef* = 0.02 to foster exploration, while the value-function loss weight was kept at $vf_{\text{coef}} = 0.5$ to balance actor and critic objectives. Gradients were clipped at *max_grad_norm* = 1 to avoid destabilising updates. Optimisation employed Adam with a learning rate of 5×10^{-4} and $\varepsilon = 10^{-7}$. Training ran for 8.0×10^{7} time-steps, with evaluation and logging performed every 1 000 steps. The hyper-parameters of the A2C agent were tuned in two sequential stages.

1. **Initialisation from established practice.** Default settings recommended by Stable-Baselines3 for long-horizon, discrete-action problems were adopted as the starting point: $\gamma = 0.99,\ n_{\text{steps}} = 128,\ \text{ent\_coef} = 0.01,\ vf_{\text{coef}} = 0.5,\ \text{max\_grad\_norm} = 0.5,\ \text{lr} = 3 \times 10^{-4}$.
2. **Coarse grid search.** A grid search comprising 20 distinct hyper-parameter combinations (five random seeds each) was run for 5 M time-steps per combination. The following ranges were explored while all other knobs were kept at the Stage-1 values:

$$\gamma \in \{0.95,\ 0.99,\ 0.995,\ 0.999\},$$
$$\text{ent\_coef} \in \{0,\ 0.01,\ 0.02,\ 0.05\},$$
$$\text{lr} \in \{1 \times 10^{-3},\ 5 \times 10^{-4},\ 1 \times 10^{-4}\}.$$

Performance was measured by mean episodic reward on a held-out validation set of 500 scenarios; the configuration with the highest score was selected for full training.

The search identified the following setting as optimal: $\gamma = 0.995$ (indicating a strong preference for delayed rewards), $n_{\text{steps}} = 128$ (balancing Generalised Advantage Estimation accuracy against memory use), $\text{ent\_coef} = 0.02$ (yielding an $\approx 3\%$ macro-F_1 gain over 0.01 and 0.05 by promoting exploration), $vf_{\text{coef}} = 0.5$ (higher values slowed policy learning with no accuracy benefits), $\text{max\_grad\_norm} = 1$ (clipping at 0.5 caused under-fitting; unclipped runs diverged), and $\text{lr} = 5 \times 10^{-4}$ with Adam ($\varepsilon = 10^{-7}$).

The fixed hyper-parameter set was subsequently trained for the full 80M time-steps, with evaluation and logging performed every 1000 steps. The RL decision process at each Level was modelled as a finite Markov Decision Process $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$:

Task	Action space (discrete)	Actions
Level-1	3	0, 1, 2
Level-2	5	0, 1, 2, 3, 4
Level-3	3	0, 1, 2
Level-4	3	0, 1, 2
Level-5	3	0, 1, 2

Table 4. Action spaces and corresponding actions for each level.

- **State** $s_t = [c_1, c_2, c_3, c_4,\ l_1, \ldots, l_5,\ q_t] \in \mathcal{S} \subset \mathbb{R}^{10}$ combines three elements

  (i) the *prediction–confidence vector* c_t (padded to length 4);
  (ii) the one-hot *level indicator* l_t (length 5);
  (iii) the scalar *credit counter* $q_t \in \{0, \ldots, 5\}$ that records how many "gather–data" credits remain at the current level.

A concrete example for Level-2 might be $s_t = [0.71, 0.22, 0.05, 0.00,\ 0, 1, 0, 0, 0,\ 4]$, meaning the `Enabler` is moderately confident the post concerns *infrastructure/utility damage* (class-1), we are at Level-2, and 4 credits are still available.

- **Action** $a_t \in \mathcal{A}(\mathrm{l}_t)$ comes from the discrete sets in Table 4; the action space therefore changes deterministically with the current level.
- **Transition** $T(s_{t+1} \mid s_t, a_t)$ is deterministic:
  - if a_t is a classification label and *correct*, advance to the next level and reset $q_{t+1} = 5$;
  - if a_t is *incorrect*, the episode terminates;
  - if a_t is `Gather Additional Data`, stay at the same level, decrement $q_{t+1} = q_t - 1$, and refresh c_{t+1} with a new sample.
- **Reward** is a scalar

$$R(s_t, a_t) = \begin{cases} +1, & \text{correct label} \\ -5, & \text{incorrect label} \\ -1, & \texttt{Gather Additional Data} \end{cases}$$

Together with $\gamma = 0.995$, the agent is encouraged to maximise long-term accuracy while sparingly using its limited credits.

- **Episode termination** occurs when the agent either (i) finishes Level-5, (ii) makes an incorrect decision, or (iii) exhausts all five credits without making a final classification.

Assume the agent is at Level-3 with $q_t = 2$ credits left and receives $\mathrm{c}_t = [0.10, 0.85, 0.05, 0.00]$. If it predicts *severe damage* ($a_t = 1$) and that is indeed the ground truth, it obtains $R = +1$ and transitions to Level-4 with $q_{t+1} = 5$. If instead it chooses `Gather Additional Data` ($a_t = 2$), it pays $R = -1$, stays at Level-3 with $q_{t+1} = 1$, and a fresh image–text pair is sampled.

Human operator decision maker

This section discusses the human operator `Decision Maker` agent, which determines the activities to be undertaken at each `Level` in a `Scenario`. The human operators are the participants who were recruited for the evaluation study. A web application called `Disaster Maestro` (https://disaster-maestro.web.app/) was developed to collect human responses and evaluate their performance in comparison to the previous reinforcement learning agent. Unlike the RL agent, human operators were not provided with judgment insights from the `Enabler` agents, as seen in Fig. 1. Instead, they had to independently assess and make decisions, simulating the decision-making process of a stakeholder in disaster management. This design aimed to replicate real-world conditions where stakeholders must often rely on their own judgment without the aid of advanced analytical tools.

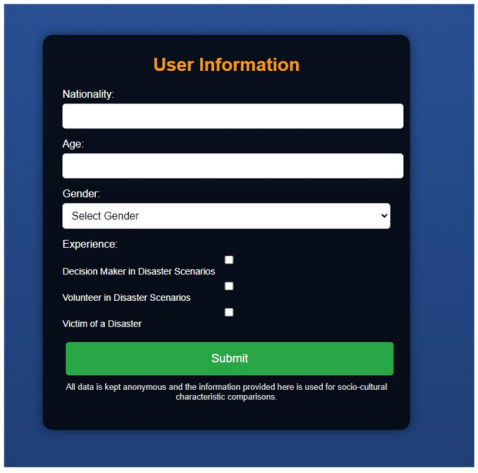
To ensure a robust evaluation of the RL agent, responses were gathered from a diverse group of participants, as per the following inclusion criteria: victims affected by disaster, volunteers involved in rescue or relief operations, or stakeholders in disaster management. Participants were recruited via networking platforms such as Facebook and LinkedIn, as well as through close contacts. The recruitment process involved searching hashtags related to disasters on social media and reaching out to individuals who had posted relevant content. The individuals were victims/volunteers/stakeholders in the 2018 floods in Kerala, typhoon Gaemi in Taiwan, and 2024 Hualien earthquake. This diverse input brought significant depth and realism to the evaluation process. The participants were provided with the URL to the web application, and asked to complete at least 2 `Scenarios` before they chose to end the survey.

- In `Level-1`, `Level-2`, and `Level-3` participants are presented with image-text pairs from the preprocessed CrisisMMD dataset and are asked to make a judgment based on the data. They then indicate their decision by clicking the button that corresponds to their decision.
- In `Level-4` participants are presented with images captured by satellites from the preprocessed xBD dataset and are asked to make a judgment based on the data. They then indicate their decision by clicking the button that corresponds to their decision.
- In `Level-5` participants are presented with images captured by drones from the preprocessed RescueNet dataset and are asked to make a judgment based on the data. They then indicate their decision by clicking the button that corresponds to their decision.

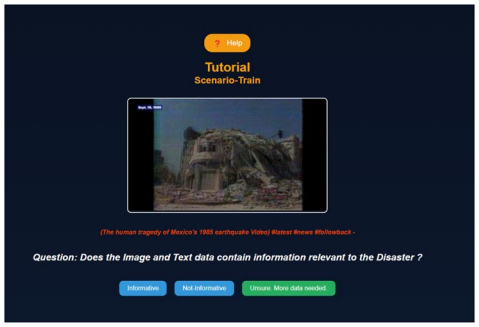
The data prepared for training and evaluating the RL agent is uploaded to a cloud storage to be utilized for the web application. The web application was carefully structured to guide users, as seen in Fig. 6.

The process begins with an introduction to the project, providing context and explaining the purpose of the exercise. Following this, a training section allows users to interact with the system and learn how to make informed decisions before they are scored. This section is crucial in helping participants understand the complexities of disaster management and the types of decisions they would need to make. In the application logic, users are presented with a series of N `Scenarios`, the `Scenarios` shown in the application come from the validation set, so that there is consistency when evaluating the human responses with the RL agent. The application computes the participants' final score and showcases insights on the correctness of their decisions. No personal data was collected from the participants of this study. In the web application, the participants are shown:

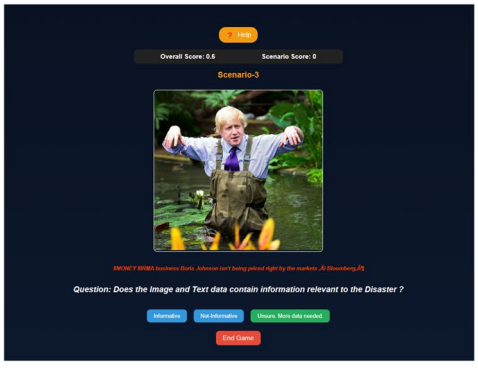
- For `Level-1`, data which can be 'informative' or 'not informative', where they inherently judge the data and then select an action 'informative', 'not informative', or `Gather Additional Data`.
- For `Level-2`, data which can represent 'affected individuals', 'infrastructure and utility damage', 'other relevant information', or 'rescue and volunteering efforts', where they judge the data and then select an action: 'affected individuals', 'infrastructure and utility damage', 'other relevant information', 'rescue and volunteering efforts', or `Gather Additional Data`.
- For `Level-3`, data which can indicate 'little or no damage' or 'severe damage', where they judge the extent of damage and then select an action: 'little or no damage', 'severe damage', or `Gather Additional Data`.
- For `Level-4`, data which can indicate 'no damage' or 'major damage', where they judge the extent of damage and then select an action: 'no damage', 'major damage', or `Gather Additional Data`.
- For `Level-5`, data which can indicate 'building no damage' or 'building destroyed', where they judge the extent of damage and then select an action: 'building no damage', 'building destroyed', or `Gather Additional Data`.



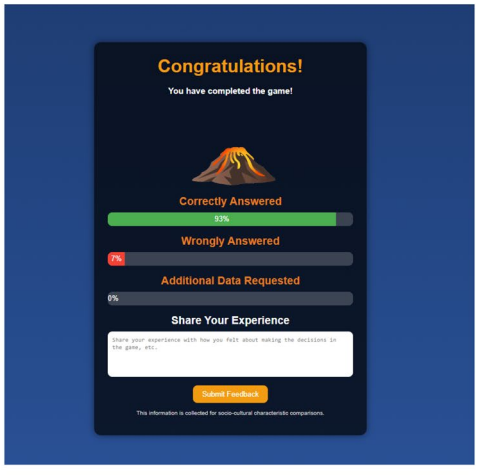
(a) GUI displaying the user form for information collection.



(b) GUI showing tutorial screen.



(c) Example GUI screen with scoring.



(d) Example GUI of results screen.

Figure 6. Example GUI screens of the web application.

The web application is built in ReactJS and hosted on Firebase, making use of the database and cloud storage to store the responses from the participants, and to store the data records to display the questions to the user in the survey. In a `Scenario`, each `Level` has an allocation of 5 credits to request additional data, requesting additional data can result in a minor penalty of−`1`. Correct decisions result in a reward of `+1` and wrong decisions result in a penalty of−`5`.

Decision maker metrics

To evaluate the performance of the `Decision Maker` agents in both cases, reinforcement learning or human operator, the following metrics were calculated:

- `tree_score`: The tree score is the total reward accumulated in one episode (one `Scenario`). The score has a range of [-5,+5], where a score of +5 indicates that the `Decision Maker` agent made correct decisions at all `Levels` within that `Scenario`. This metric quantifies how well the `Decision Maker` agent performs across different `Scenario`.
- `isTreeCorrectlyAnswered`: This metric is calculated as the mean number of correct decisions made at each `Level` in a `Scenario`, providing a confidence score of the `Decision Maker` agent's performance. A value of 1.0 indicates that the agent made correct decisions at all `Levels` in a `Scenario`.
- `isTreeWronglyAnswered`: This metric is calculated by the mean number of wrong decisions made at each `Level` in a `Scenario`, providing a confidence score of the `Decision Maker` agent's performance. A value of 1.0 indicates that the agent made wrong decisions at all `Levels` in a `Scenario`.
- `isGatherAdditionalDataRequested`: This metric is calculated by the mean number of times the gather additional data action was utilized at each `Level` in a `Scenario`, providing an analysis of the `Decision Maker` agent's performance. A value of 1.0 indicates that the agent requested additional data at all `Levels` in a `Scenario`.

For all the metrics, the mean and standard deviation are calculated at regular intervals, with an interval of 1000 steps.

Results

This section presents our study's experimental results. First, a brief overview of the `Enabler` agents' results is provided, followed by the core of the results which is the comparison of the performance of the autonomous `Decision Maker` agent to that of a human operator.

Enabler agent

This section discusses the experimental results for the `Enabler` agents for each `Level` in a `Scenario`, using the metrics introduced in Sect. 4.2.

Level-1

In `Level`-1, the `Enabler` agent judges the image + text data to verify its relevance in relation to the disaster, as seen in Fig. 7, classifying the results into 2 classes: 'informative' and 'not informative'. The performance metrics for the best model are shown in Table 5. From the table, a Macro Average F1-score of 82.14% indicates that the model's performance is balanced across both classes.

Level-2

In `Level`-2, the `Enabler` agent evaluates image-text data to verify its relevance to disaster aid, as illustrated in Fig. 8. The data is classified into four categories: 'affected individuals', 'infrastructure and utility damage', 'other relevant information', and 'rescue, volunteering, or donation efforts'. The performance metrics for the best model are presented in Table 6. The model achieves a moderate Macro Average F1-score of 74.36%, indicating reasonably balanced performance across the four classes. Notably, the model performs best on 'other relevant information' with an F1-score of 89.83%, and 'infrastructure and utility damage' with an F1-score of 87.02%, while the performance on 'affected individuals' is lower, with an F1-score of 48.78%, but this is expected due to lower of number of samples available to train for the 'affected individuals' class.

Level-3

In `Level`-3, the `Enabler` agent judges the image + text data to verify its disaster damage captured by victims in relation to the disaster, as seen in Fig. 9. The model is trained for 2 classes: 'little or no damage' and 'severe damage'. The performance metrics for the best model are shown in Table 7. From the table, a Macro Average F1-score of 73.43% indicates reasonably balanced performance across both classes.

Level-4

In `Level`-4, the `Enabler` agent judges the image to verify its disaster damage captured by satellites in relation to the disaster, as seen in Fig. 10. The classification model is trained for 2 classes: 'no damage' and 'major damage'. The performance metrics for the best model are shown in Table 8. From the table, a Macro Average F1-score of 99.89% indicates the model's performance is balanced across both classes. The model correctly identified 99.38% of 'major damage' instances with a precision of 99.85%. Similarly, the model achieved 100% recall for 'no damage' instances with a precision of 100%.

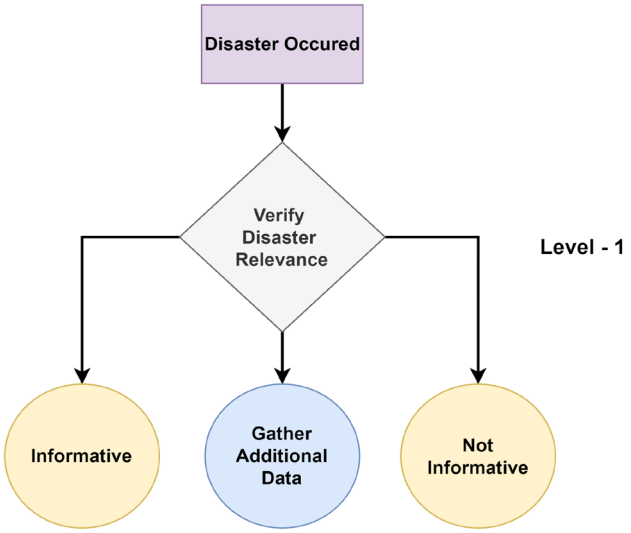


Figure 7. Overview of level-1.

Label	Precision	Recall	F1-score	Support
Informative (0)	0.8665	0.7730	0.8171	1855
Not informative (1)	0.7832	0.8731	0.8257	1742
Accuracy	0.8215			
Macro avg	0.8248	0.8231	0.8214	3597
Weighted avg	0.8261	0.8215	0.8213	3597

Table 5. Performance metrics for the level-1 enabler agent.

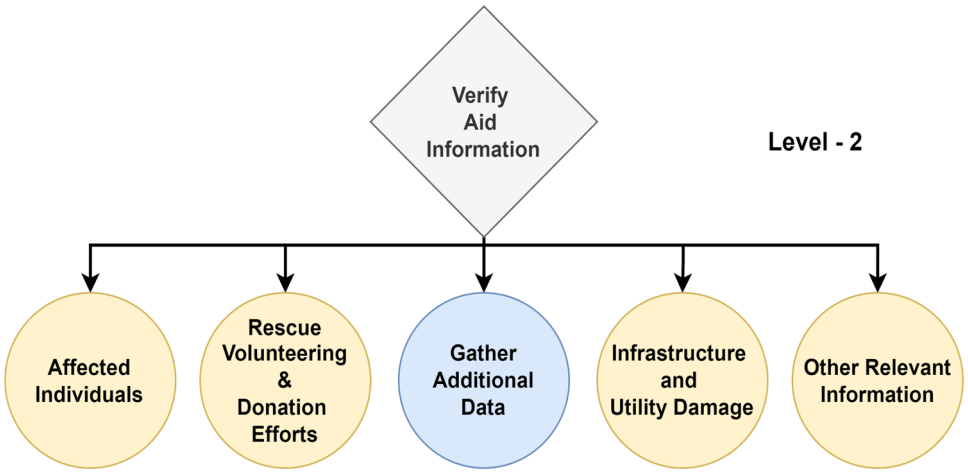


Figure 8. Overview of level-2.

Level-5

In `Level-5`, the `Enabler` agent judges the image to verify its disaster damage captured by drones in relation to the disaster, as seen in Fig. 11. The classification model is trained for 2 classes: 'building no damage' and 'building destroyed'. The performance metrics for the best model are shown in Table 9. From the table, a Macro Average F1-score of 65.28% indicates reasonably balanced performance across both classes. The model correctly identified 72.44% of 'building no damage' instances with a precision of 71.76%. Similarly, the model achieved 47.24% recall for 'building destroyed' instances with a precision of 71.76%, the model finds it hard to find all instances of building destroyed, which can be attributed to the similar features between the classes, resulting in confusion among the classes.

Label	Precision	Recall	F1-score	Support
Affected individuals (0)	0.4667	0.5109	0.4878	137
Infrastructure and utility damage (1)	0.8486	0.8930	0.8702	766
Other relevant information (2)	0.9158	0.8814	0.8983	506
Rescue/volunteering/donation effort (3)	0.7476	0.6906	0.7179	446
Accuracy	0.8129			
Macro avg	0.7447	0.7440	0.7436	1855
Weighted avg	0.8145	0.8129	0.8130	1855

Table 6. Performance metrics for the Level-2 enabler agent.

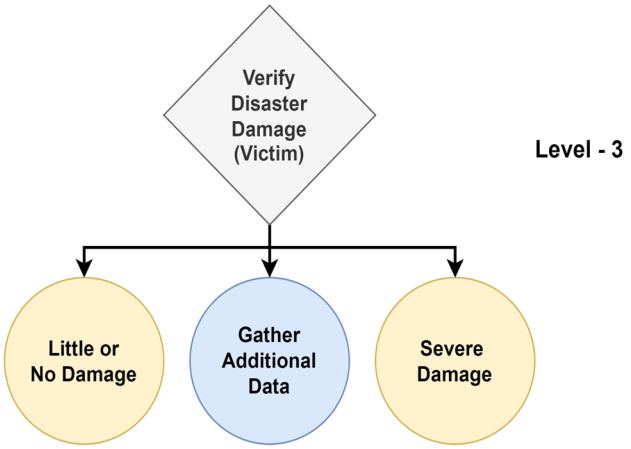


Figure 9. Overview of level-3.

Label	Precision	Recall	F1-Score	Support
Little or no damage (0)	0.6438	0.7148	0.6775	263
Severe damage (1)	0.8188	0.7652	0.7911	443
Accuracy	0.7465			
Macro avg	0.7313	0.7400	0.7343	706
Weighted avg	0.7536	0.7465	0.7488	706

Table 7. Performance metrics for the Level-3 Enabler Agent.

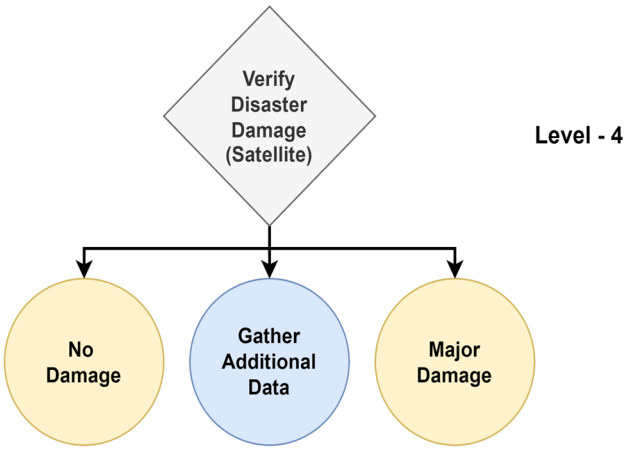


Figure 10. Overview of level-4.

Label	Precision	Recall	F1-score	Support
No damage	1.0000	1.0000	1.0000	29856
Major damage	0.9985	0.9938	0.9961	12355
Micro avg	0.9996	0.9982	0.9989	42211
Macro avg	0.9993	0.9969	0.9981	42211
Weighted avg	0.9996	0.9982	0.9989	42211

Table 8. Performance metrics for Level-4 enabler agent.

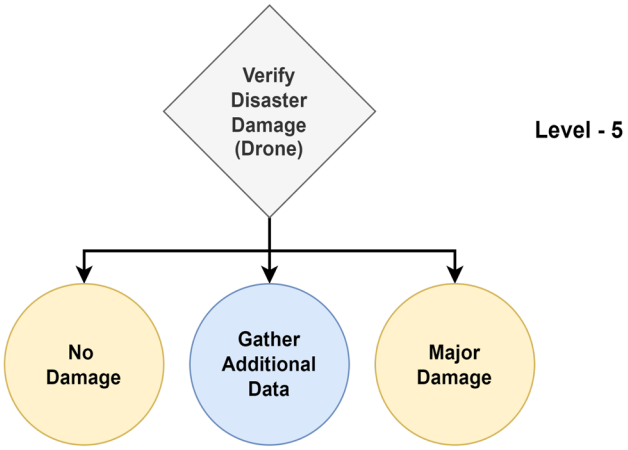


Figure 11. Overview of level-5.

Label	Precision	Recall	F1-score	Support
Building no damage	0.7475	0.7244	0.7358	1974
Building destroyed	0.7176	0.4724	0.5698	1232
Micro avg	0.7386	0.6276	0.6786	3206
Macro avg	0.7326	0.5984	0.6528	3206
Weighted avg	0.7360	0.6276	0.6720	3206

Table 9. Performance metrics for Level-5 enabler agent.

Enabler agent	Macro avg. F1-score
Level-1	0.8214
Level-2	0.7436
Level-3	0.7343
Level-4	0.9989
Level-5	0.6528

Table 10. Results summary for the enabler agents trained for each level in a scenario.

Results summary for enabler agents

The performance of the `Enabler` agents across different levels in the structured decision-making framework for disaster management varies, as reflected by the Macro Average F1-Scores in Table 10. `Level-1`, responsible for classifying data as informative or not, achieved a balanced performance with an F1-Score of 82.14%. `Level-2`, which identifies relevant humanitarian efforts, showed moderate performance with an F1-Score of 74.36%. `Level-3`, focused on assessing damage severity based on victim-captured data, attained a reasonably balanced F1-Score of 73.43%. The highest performance was observed in `Level-4`, where satellite imagery was used to distinguish between major and no damage, achieving an exceptional F1-Score of 99.89%. In contrast, `Level-5`, which deals with drone-captured data, exhibited the lowest performance with an F1-Score of 65.28%, indicating challenges in distinguishing between undamaged and destroyed buildings. Since the primary objective of the framework is to demonstrate autonomous decision-making rather than the performance of the `Enabler`

agents, the performance is satisfactory. It effectively provides judgment insights on the data to inform the RL Decision Maker agent.

Decision maker agent

This section presents the experimental results from training and evaluating the Decision Maker agents across multiple Scenarios, using the metrics introduced in "Decision maker metrics". The metric plots are generated with a 10% running average smoothing to highlight learning trends.

Reinforcement learning

An A2C reinforcement learning algorithm is trained across multiple Scenarios, where each episode represents a Scenario, and each step corresponds to a Level. To evaluate how well the RL agent can perform and what the limit to performance is by relying solely on the outputs from the Enabler agents (judgement insights), an Benchmark agent is implemented. The Benchmark serves as a benchmark agent, navigating all Scenarios in validation/training by selecting actions based solely on the index of the maximum value in the prediction score array from the judgment data (Enabler agent output). This benchmark is used to assess how decision-making based solely on judgment insights compares to the RL agent. Additionally, it helps define the stopping criteria for the RL agent's learning process. Performance plots for the RL agent and the Benchmark are shown in Figs. 12 and 13 respectively.

From the performance plots of the Benchmark and the RL agent, it is seen that relying solely on judgment insights leads to uncertain decision-making, as evidenced by the standard deviation of the 'tree_score' for the Benchmark being 11.5218 (Fig. 13d)—approximately 3 times larger than that of the RL agent, which was 4.49 (Fig. 12a).

The mean value of 'isTreeCorrectlyAnswered' for the Benchmark indicated that using only judgment insights resulted in an accuracy of 82.01% (Fig. 13a) across all validation Scenarios, whereas the RL agent achieved 88% accuracy (Fig. 12c). The stopping criteria for the RL agent were set to trigger when the mean 'isTreeCorrectlyAnswered' exceeded the Benchmark's mean of 88% (Fig. 13a) on the training dataset and when the mean 'tree_score' surpassed the Benchmark's mean of 1.5 (Fig. 13c). According to these results, the RL agent outperformed the Benchmark agent, with a mean accuracy 7.32% larger than the Benchmark agent.

The standard deviation of this metric was also strongly reduced, from 0.3841 for the Benchmark agent to 0.15 for the RL agent, showing a 60.94% decrease.

The training and evaluation plots of the RL agent, shown in Fig. 12, clearly illustrate the agent's learning progression. For the RL agent, an evaluation on 1809 Scenarios from the validation data revealed that 1736 Scenarios achieved a perfect 'tree_score' of 5, indicating correct decisions at all Levels for those Scenarios. In these cases, the 'isTreeWronglyAnswered' metric had a value of 0.1, meaning the agent made incorrect decisions only 10% of the time. For Scenarios with 'tree_scores' of 2, 3, and 4, the lower scores were due to the agent requesting additional data.

Initially, the RL agent made incorrect decisions 55.2% of the time (Fig. 12g) and requested additional data 20.44% of the time (Fig. 12e) on evaluation data. However, as training progressed, the agent significantly improved, reducing incorrect decisions to 12%. During evaluation, the agent correctly answered all Levels in a Scenario 88% of the time without requesting additional data, achieving a mean 'tree_score' of 1.4 (Fig. 12a) which is 1.39 times higher than the Benchmark's mean tree score. This demonstrates that the RL agent significantly enhances decision-making compared to relying solely on judgment data.

The summary of the most relevant results can be seen in Table 11.

Human operator

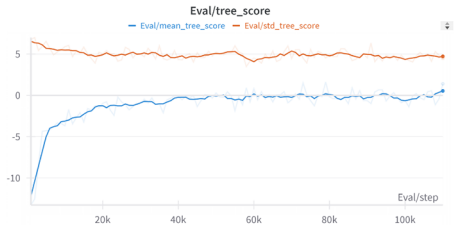
For the human evaluation study, a total of 61 responses were recorded from participants who completed the survey on the web application, as seen in Table 12. Among these, 12 were stakeholders involved in rescue or relief operations, 19 were victims affected by the disaster, and 30 were volunteers engaged in related activities. Recruiting stakeholders proved challenging due to waiting periods and their limited availability, while victims and volunteers were more willing to participate and found the survey informative.

Plots of metrics computed from individual participant responses were generated to highlight trends, and are shown in Fig. 14.

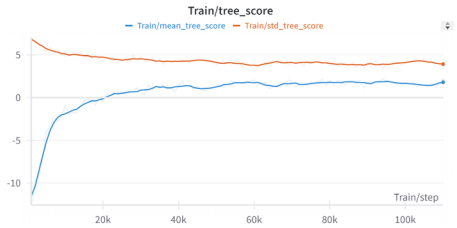
Participants in the victim category completed an average of 10.47 Scenarios, compared to 6.67 by stakeholders and 5.25 by volunteers. Cumulatively, victims completed 199 Scenarios, while stakeholders and volunteers completed 80 and 142, respectively. This resulted in victims achieving a mean 'tree_score' of − 0.88, which was better than the − 1.08 and − 1.18 scores from stakeholders and volunteers, respectively (Fig. 14d). The victims' mean 'tree_score' is approximately 1.23 times better than that of stakeholders and 1.34 times better than that of volunteers.

The difference in scores can be partly attributed to the fact that stakeholders and volunteers were more attentive to detail and less likely to make hasty decisions. This is evidenced by their higher rates of requesting additional data (Fig. 14a): 11.64% for stakeholders and 11.63% for volunteers, compared to 9% for victims.

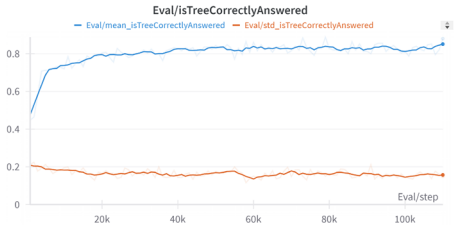
When assessing decision accuracy across the Scenarios, all participants performed similarly: stakeholders made correct decisions 65.22% of the time (Fig. 14c) and incorrect decisions 32.63% of the time (Fig. 14b); volunteers made correct decisions 61.30% of the time and incorrect decisions 34.75%; and victims made correct decisions 65.80% of the time and incorrect decisions 30.52%. The plot reveals that, although more responses were recorded from volunteers than victims, victims achieved higher 'tree_scores'. This improvement is attributed to victims completing significantly more Scenarios on average compared to both volunteers and stakeholders.



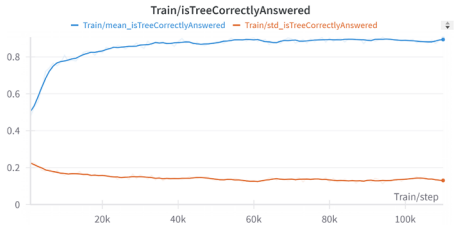
(a) (Eval): μ=1.4 and σ=4.49 of 'tree_score'



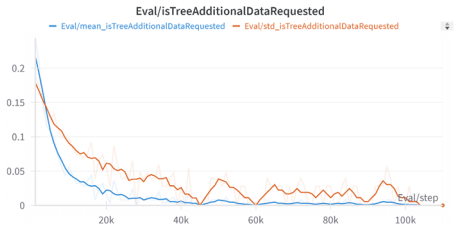
(b) (Train): μ=2.05 and σ=3.84 of 'tree_score'



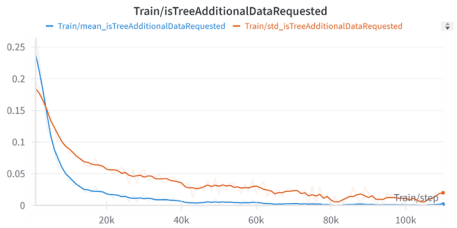
(c) (Eval): μ=0.88 and σ=0.15 of 'isTreeCorrectlyAnswered'



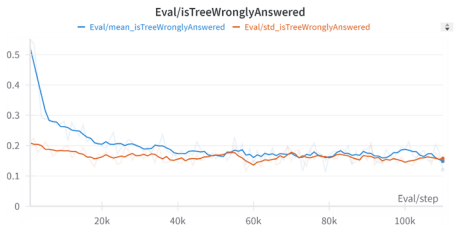
(d) (Train): μ=0.90 and σ=0.13 of 'isTreeCorrectlyAnswered'



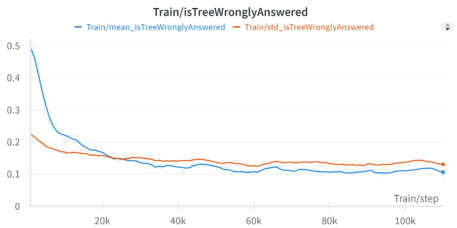
(e) (Eval): μ=0 and σ=0 of 'isAdditionalDataRequested'



(f) (Train): μ=0.01 and σ=0 of 'isAdditionalDataRequested'



(g) (Eval): μ=0.12 and σ=0.14 of 'isTreeWronglyAnswered'



(h) (Train): μ=0.12 and σ=0.09 of 'isTreeWronglyAnswered'

Figure 12. Training and evaluation metric plots for the RL agent using the A2C algorithm. (**a**,**b**) show the Mean and Standard Deviation of 'tree_score'. (**c**,**d**) display the Mean and Standard Deviation of 'isTreeCorrectlyAnswered'. (**e**,**f**) present the Mean and Standard Deviation of 'isTreeAdditionalDataRequested'. (**g**,**h**) show the Mean and Standard Deviation of 'isTreeWronglyAnswered'.

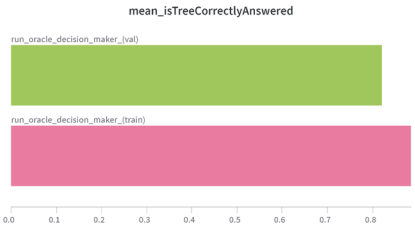
Comparative analysis

This section presents a comparative analysis of the human evaluation study results versus the reinforcement learning (RL) decision maker.

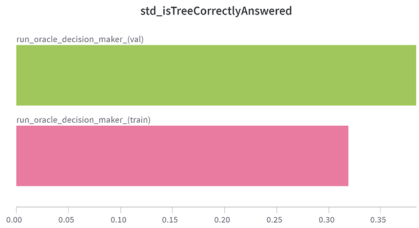
RL vs stakeholder

This section provides a comparative analysis of the aggregate performance of all stakeholders and the reinforcement learning agent. Additionally, comparisons are made with the stakeholder who completed the most `Scenarios`, against the RL agent.

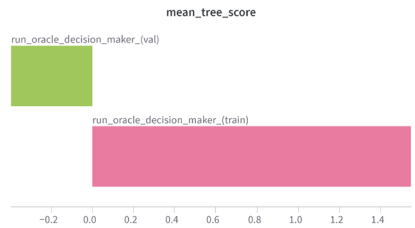
From Table 13, it is evident that the reinforcement learning (RL) agent outperforms the aggregate of stakeholder responses across all metrics. The RL agent achieves a Mean Tree Score (M.T.S) of 1.4, significantly higher than the stakeholder's score of – 1.0082. In terms of accuracy, the RL agent makes correct decisions 88% of the time (M.C.A = 0.88), whereas stakeholders collectively achieve a lower accuracy of 65.22% (M.C.A = 0.6522). Furthermore, the RL agent makes fewer incorrect decisions, with a Mean 'isTreeWronglyAnswered' (M.W.A) of



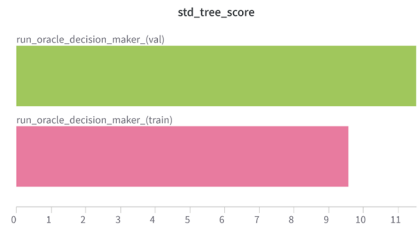
(a) μ_{val}=0.8201 and μ_{train}=0.8850 for 'isTreeCorrectlyAnswered'



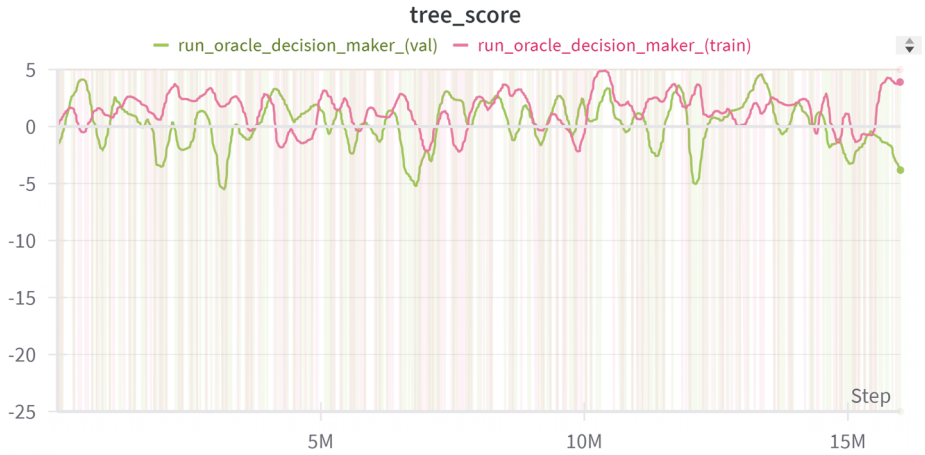
(b) σ_{val}=0.3841 and σ_{train}=0.3191 for 'isTreeCorrectlyAnswered'



(c) μ_{val}=-0.3954 and μ_{train}=1.5491 for 'tree_score'



(d) σ_{val}=11.5218 and σ_{train}=9.5717 for 'tree_score'



(e) 'tree_score'

Figure 13. Training and evaluation metric plots for the Benchmark agents algorithm. (**a**,**b**) show the Mean and Standard Deviation of 'isTreeCorrectlyAnswered'. (**c**,**d**) display the Mean and Standard Deviation of 'tree_score'. (**e**,**f**) present the 'tree_score' logged for all scenarios with 5% Gaussian smoothing applied to visualize the trend.

Metric	Type	μ	σ
tree_score	RL	1.4	4.49
	Benchmark	− 0.3954	11.5218
isTreeCorrectlyAnswered	RL	0.88	0.15
	Benchmark	0.8201	0.3841

Table 11. Summary of the performance of the RL and benchmark agent showing the mean (μ) and standard deviation (σ) for 'tree_score' and 'isTreeCorrectlyAnswered' metrics.

12%, which is approximately 0.37 times smaller than the stakeholder's M.W.A of 32.63%. Finally, the RL agent does not request additional data at all (M.A.D = 0), in contrast to stakeholders, who request additional data 11.64% of the time (M.A.D = 0.1164). Overall, the RL agent demonstrates superior performance in decision-making, accuracy, and efficiency compared to the collective stakeholder responses.

Role	Indian	Japanese	American	Canadian	British	Taiwanese
Stakeholder	10	0	0	0	1	1
Volunteer	27	0	1	0	1	1
Victim	15	1	0	1	1	1

Table 12. Demographic data from the human evaluation study.

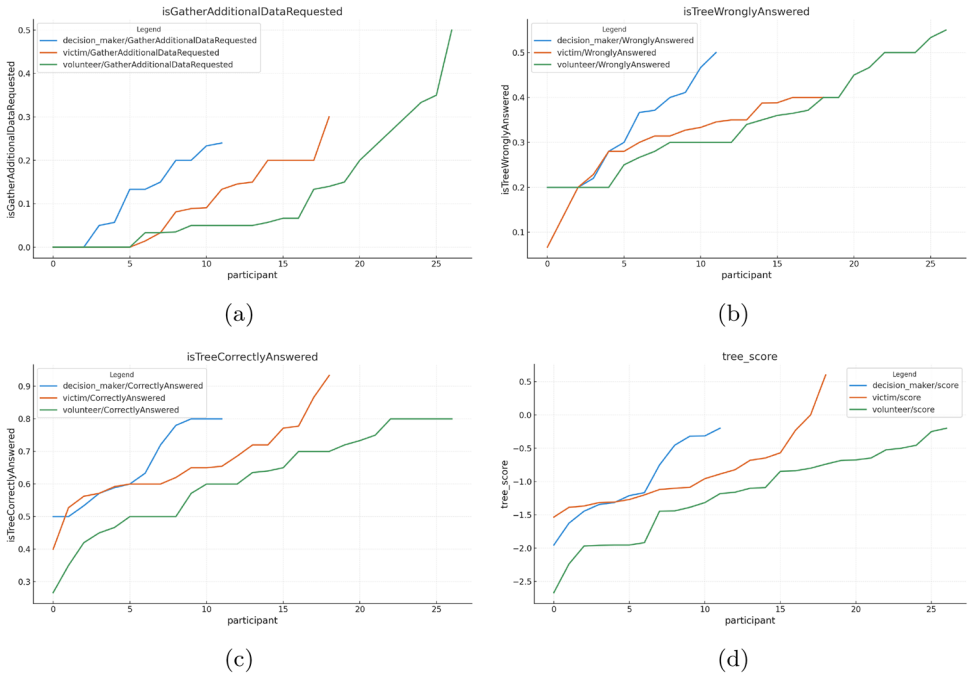


Figure 14. Metrics for the individual responses from the Human Operators in the survey. The horizontal axes represent each of the individual participants. (**a**) 'isTreeAdditionalDataRequested', (**b**) 'isTreeWronglyAnswered', (**c**) 'isTreeCorrectlyAnswered', and (**d**) 'tree_score'.

Agent	M.T.S	M.C.A	M.W.A	M.A.D
Stakeholder A (collective)	− 1.0082	0.6522	0.3263	0.1164
Stakeholder B (most scenarios)	− 1.3443	0.5889	0.4111	0.1333
RL agent	**1.4000**	**0.8800**	**0.1200**	**0.0000**

Table 13. Results comparing the RL agent with the Stakeholder participant responses: Stakeholder A (Aggregate of all stakeholder responses), and Stakeholder B (Most Scenarios Completed), showing: Mean Tree Score (M.T.S), Mean 'isTreeCorrectlyAnswered' (M.C.A), Mean 'isTreeWronglyAnswered' (M.W.A), and Mean 'isGatherAdditionalDataRequested' (M.A.D).

Additionally, from Table 13, an analysis between the RL agent and Stakeholder B (the stakeholder participant who completed 18 `Scenarios`) reveals that stakeholder tends to make less accurate decisions as they complete more `Scenarios`, only 58.89% accuracy, across the 18 `Scenarios` they completed, whereas the RL agent achieves 88% accuracy across the validation data, which is 1809 `Scenarios`. The observed trend of stakeholders making more inaccurate decisions as they complete additional `Scenarios` can be attributed to community disaster fatigue, particularly the sense of defeatism. As highlighted in[22], prolonged exposure to disaster-related decision-making often results in decreased accuracy and effectiveness among stakeholders.

RL vs volunteer

This section provides a comparative analysis between the aggregate performance of all volunteers and the reinforcement learning agent. Additionally, comparisons are made with the volunteer participant who completed the most `Scenarios`, against the RL agent.

From Table 14, it is evident that the reinforcement learning (RL) agent outperforms the aggregate of volunteer responses across all metrics. The RL agent achieves a Mean Tree Score (M.T.S) of 1.4, significantly higher than the

Agent	M.T.S	M.C.A	M.W.A	M.A.D
Volunteer A (collective)	− 1.1825	0.6131	0.3475	0.1166
Volunteer B (most scenarios)	− 1.1818	0.6353	0.3647	0.0353
RL agent	**1.4000**	**0.8800**	**0.1200**	**0.0000**

Table 14. Results comparing the RL agent with the Volunteer participant responses: Volunteer A (Aggregate of all stakeholder responses), and Volunteer B (Most Scenarios Completed), showing: Mean Tree Score (M.T.S), Mean 'isTreeCorrectlyAnswered' (M.C.A), Mean 'isTreeWronglyAnswered' (M.W.A), and Mean 'isGatherAdditionalDataRequested' (M.A.D).

Agent	M.T.S	M.C.A	M.W.A	M.A.D
Victim A (collective)	− 0.8835	0.6580	0.3052	0.0967
Victim B (most scenarios)	− 1.3077	0.5922	0.3882	0.0000
RL agent	**1.4000**	**0.8800**	**0.1200**	**0.0000**

Table 15. Results comparing the RL agent with the Victim participant responses: Victim A (Aggregate of all stakeholder responses), and Victim B (Most Scenarios Completed), showing: Mean Tree Score (M.T.S), Mean 'isTreeCorrectlyAnswered' (M.C.A), Mean 'isTreeWronglyAnswered' (M.W.A), and Mean 'isGatherAdditionalDataRequested' (M.A.D).

volunteer's score of − 1.1825. In terms of accuracy, the RL agent makes correct decisions 88% of the time (M.C.A = 0.88), whereas volunteers collectively achieve a lower accuracy of 61.31% (M.C.A = 0.6131). Furthermore, the RL agent makes fewer incorrect decisions, with a Mean 'isTreeWronglyAnswered' (M.W.A) of 12%, which is approximately 0.35 times smaller than the volunteers' M.W.A of 34.75%. Finally, the RL agent does not request additional data at all (M.A.D = 0), in contrast to volunteers, who request additional data 11.66% of the time (M.A.D = 0.1166). Overall, the RL agent demonstrates superior performance in decision-making, accuracy, and efficiency compared to the collective volunteer responses.

Additionally, from Table 14, an analysis between the RL agent and Volunteer B (the volunteer participant who completed 17 `Scenarios`) reveals that volunteers tend to make less accurate decisions as they complete more `Scenarios`, only 63.53% accuracy, across the 17 `Scenarios` they completed, whereas the RL agent achieves 88% accuracy across the validation data, which is 1809 `Scenarios`. The observed trend of volunteers making more inaccurate decisions as they complete additional `Scenarios` can also be attributed to community disaster fatigue, similar to the observed trend seen for stakeholder responses in "RL vs stakeholder".

RL vs victim

This section offers a comparative analysis between the aggregate performance of all victim participants and the reinforcement learning (RL) agent. Additionally, it compares the RL agent with the victim participant who completed the most `Scenarios`. Although victims typically do not participate in decision-making processes within disaster management, this analysis provides insights into how individuals from different backgrounds perform against an RL agent in disaster-related decision-making. The comparison is based on the learning curve of the victims in understanding disaster-related decision-making using the information and tools provided to the victim participants through the web application.

From Table 15, it is evident that the reinforcement learning (RL) agent outperforms the aggregate of victim responses across all metrics. The RL agent achieves a Mean Tree Score (M.T.S) of 1.4, significantly higher than the victim's score of -0.8835. In terms of accuracy, the RL agent makes correct decisions 88% of the time (M.C.A = 0.88), whereas victims collectively achieve a lower accuracy of 65.80% (M.C.A = 0.6580). Furthermore, the RL agent makes fewer incorrect decisions, with a Mean 'isTreeWronglyAnswered' (M.W.A) of 12%, which is approximately 0.39 times smaller than the victims's M.W.A of 30.52%. Finally, the RL agent does not request additional data at all (M.A.D = 0), in contrast to victims, who request additional data 9.67% of the time (M.A.D = 0.0967). Overall, the RL agent demonstrates superior performance in decision-making, accuracy, and efficiency compared to the collective volunteer responses.

Additionally, from Table 15, an analysis between the RL agent and Victim B (the victim participant who completed 51 `Scenarios`) reveals that victims tend to make less accurate decisions as they complete more `Scenarios`, only 59.22% accuracy which is almost random decisions, across the 51 `Scenarios` they completed, whereas the RL agent achieves 88% accuracy across the validation data, which is 1809 `Scenarios`. Participants identified as victims tended to complete more `Scenarios`, but this did not translate into more accurate decision-making. Their decisions appeared almost random across the `Scenarios`.

RL vs All

This section offers a comparative analysis between the aggregate performance of all participants, independent of their role, and the reinforcement learning (RL) agent.

From Table 16, it is evident that the reinforcement learning (RL) agent outperforms the aggregate of all responses across all metrics. The RL agent achieves a Mean Tree Score (M.T.S) of 1.4, significantly higher than

Agent	M.T.S	M.C.A	M.W.A	M.A.D
All (collective)	− 1.0651	0.6334	0.3301	0.1042
RL agent	**1.4000**	**0.8800**	**0.1200**	**0.0000**

Table 16. Results comparing the RL agent with the all participant responses showing: Mean Tree Score (M.T.S), Mean 'isTreeCorrectlyAnswered' (M.C.A), Mean 'isTreeWronglyAnswered' (M.W.A), and Mean 'isGatherAdditionalDataRequested' (M.A.D).

the collective participant score of − 1.0651. In terms of accuracy, the RL agent makes correct decisions 88% of the time (M.C.A = 0.88), whereas collectively the participants achieve a lower accuracy of 63.34% (M.C.A = 0.6334). According to these results, the RL agent outperformed the participants, with a mean accuracy 38.93% larger than the participants.

Furthermore, the RL agent makes fewer incorrect decisions, with a Mean 'isTreeWronglyAnswered' (M.W.A) of 12%, which is approximately 2.75 times smaller than the collective of all the participants' M.W.A (33.01%). Finally, the RL agent does not request additional data at all (M.A.D = 0), in contrast, the participants collectively request additional data 10.42% of the time (M.A.D = 0.1042). Overall, the RL agent demonstrates superior performance in decision-making, accuracy, and efficiency compared to the collective volunteer responses.

Discussions summary

From the results for the `Decision Maker` agents, it is quite evident that the proposed framework for structured decision-making, has enabled the reinforcement learning `Decision Maker` to outperform the `Benchmark` agent and the Human operators, with a mean 'tree_score' 1.39 times larger than the `Benchmark` agent and 1.62 times larger than the collective mean 'tree_score' score of the human participants from the study, irrespective of their role. In addition, the RL agent outperformed the `Benchmark` agent and the Human operators, with a mean accuracy ('isTreeCorrectlyAnswered') 7.32% larger than the `Benchmark` agent and 38.93% larger than the accuracy of decisions made across multiple scenarios by the Human operators. Also, the stability of the RL agent making consistently correct decisions is 60.94% higher than the `Benchmark` agent, which is calculated as the reduction in the standard deviations of the accuracy.

The stability in decision-making provided by this framework is critical for the safety-critical decisions required in disaster management. The performance statistics clearly demonstrate that integrating structure into autonomous decision-making can effectively make decisions more reliable and justifiable, mitigating challenges such as community fatigue, inter-agency coordination issues, and data overload faced by stakeholders. To further enhance the suitability of this framework for practical applications, incorporating validation checks for ethical and legal compliance, along with safety mechanisms for irreversible decisions, is essential.

Conclusions

This paper proposed a structured decision-making framework for autonomous decision-making in the context of disaster management. The framework introduced concepts of `Scenarios`, `Levels` and `Enabler` agents that aid the reinforcement learning `Decision Maker` agent in autonomous decision-making. The framework's performance was rigorously evaluated against systems that rely solely on judgement-based insights, `Benchmark` agents, as well as human operators who have disaster experience: victims, volunteers, and stakeholders. The results demonstrate that the structured decision-making framework achieves 60.94% greater stability in consistently accurate decisions across multiple `Scenarios`, compared to judgement-based systems. Moreover, the framework was shown to outperform human operators with a 38.93% higher accuracy across various `Scenarios`. This research highlights the potential benefits of integrating structure into safety-critical decision-making, paving the way for future improvements.

Data availability

The Code used to showcase the experiments carried out in the research is publicly available at https://github.com/From-Governance-To-Autonomous-Robots/Autonomous-Governance-in-Disaster-Management.

Received: 23 December 2024; Accepted: 6 August 2025
Published online: 01 September 2025

References

1. Raungratanaamporn, I.-S., Pakdeeburee, P., Kamiko, A. & Denpaiboon, C. Government-communities collaboration in disaster management activity: Investigation in the current flood disaster management policy in thailand. *Proc. Environ. Sci.* **20**, 658–667. https://doi.org/10.1016/j.proenv.2014.03.079 (2014).
2. Linardos, V., Drakaki, M., Tzionas, P. & Karnavas, Y. L. Machine learning in disaster management: Recent developments in methods and applications. *Mach. Learn. Knowl. Extract.* **4**(2), 446–473. https://doi.org/10.3390/make4020020 (2022).
3. Roberts, M., Apker, T., Johnson, B., Auslander, B., Wellman, B., & Aha, D.W. Coordinating robot teams for disaster relief. In: Proceedings of the Twenty-Eighth Florida Artificial Intelligence Research Society Conference. AAAI Press, Hollywood, FL (2015). An extended version of this paper appears as: Roberts, M., Vattam, S., Alford, R., Auslander, Apker, T., Johnson, B., & Aha, D.W. (2015). Goal reasoning to coordinate robotic teams for disaster relief. In A. Finzi, F. Ingrand, & Andrea Orlandini(Eds.) Planning and Robotics: Papers from the ICAPS Workshop. Jerusalem, Israel: AAAI Press. http://makro-nrl.bitbucket.io/publications/robertsEtAl15.flairs.coordinatingTeams.pdf.

4. Salvini, P. et al. Human involvement in autonomous decision-making systems lessons learned from three case studies in aviation, social care and road vehicles. *Front. Polit. Sci.* **5**, 1. https://doi.org/10.3389/fpos.2023.1238461 (2023).
5. Arifah, A., Tariq, M. & Juni, M. H. Decision making in disaster management cycle of natural disasters: A review. *Int. J. Public Health Clin. Sci.* **1**, 1 (2019).
6. Smith, W. & Dowell, J. A case study of co-ordinative decision-making in disaster management. *Ergonomics* **43**, 1153–1166 (2000).
7. Amiran, A., Behnam, B. & Seyedin, S. Ai-based model for site-selecting earthquake emergency shelters. *Sci. Rep.* **14**(1), 29033. https://doi.org/10.1038/s41598-024-80586-w (2024).
8. Shukla, D., Azad, H. K., Abhishek, K. & Shitharth, S. Disaster management ontology- an ontological approach to disaster management automation. *Sci. Rep.* **13**(1), 8091. https://doi.org/10.1038/s41598-023-34874-6 (2023).
9. Sun, L., Shawe-Taylor, J. & D'Ayala, D. Artificial intelligence-informed planning for the rapid response of hazard-impacted road networks. *Sci. Rep.* **12**(1), 16286. https://doi.org/10.1038/s41598-022-19637-z (2022).
10. Fan, Z., Lin, Y., Ai, Y. & Xu, H. Adaptive task migration strategy with delay risk control and reinforcement learning for emergency monitoring. *Sci. Rep.* **14**(1), 17606. https://doi.org/10.1038/s41598-024-67886-x (2024).
11. Gupta, T., & Roy, S. A hybrid model based on fused features for detection of natural disasters from satellite images. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 1699–1702 (2020)
12. Ofli, F., Alam, F., & Imran, M. Analysis of social media data using multimodal deep learning for disaster response. CoRR arXiv:2004.11838 (2020).
13. Yang, L. & Cervone, G. Analysis of remote sensing imagery for disaster assessment using deep learning: a case study of flooding event. *Soft. Comput.* **23**(24), 13393–13408. https://doi.org/10.1007/s00500-019-03878-8 (2019).
14. Dotel, S., Shrestha, A., Bhusal, A., Pathak, R., Shakya, A., & Panday, S.P. Disaster assessment from satellite imagery by analysing topographical features using deep learning. In *Proceedings of the 2020 2nd International Conference on Image, Video and Signal Processing*. IVSP '20, pp. 86–92. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3388818.3389160 (2020).
15. Alam, F., Ofli, F., & Imran, M. Crisismmd: Multimodal twitter datasets from natural disasters. CoRR arXiv:1805.00713 (2018).
16. Rezk, M., Elmadany, N., Hamad, R. K. & Badran, E. F. Categorizing crises from social media feeds via multimodal channel attention. *IEEE Access* **11**, 72037–72049. https://doi.org/10.1109/ACCESS.2023.3294474 (2023).
17. Gupta, R., Hosfelt, R., Sajeev, S., Patel, N., Goodman, B., Doshi, J., Heim, E.T., Choset, H., & Gaston, M.E. xbd: A dataset for assessing building damage from satellite imagery. CoRR arXiv:1911.09296 (2019).
18. Rahnemoonfar, M., Chowdhury, T., & Murphy, R. Rescuenet: A high resolution uav semantic segmentation dataset for natural disaster damage assessment. *Sci. Data* **10**(1). https://doi.org/10.1038/s41597-023-02799-4 (2023).
19. Kaur, N., Lee, C.-C., Mostafavi, A., & Mahdavi-Amiri, A. Large-scale building damage assessment using a novel hierarchical transformer architecture on satellite images. arXiv preprint arXiv:2208.02205 (2022)
20. Bharosa, N., Lee, J. & Janssen, M. Challenges and obstacles in sharing and coordinating information during multi-agency disaster response: Propositions from field exercises. *Inf. Syst. Front.* **12**, 49–65. https://doi.org/10.1007/s10796-009-9174-z (2010).
21. Misra, S., Roberts, P. & Rhodes, M. Information overload, stress, and emergency managerial thinking. *Int. J. Disaster Risk Reduction* **51**, 101762. https://doi.org/10.1016/j.ijdrr.2020.101762 (2020).
22. Ingham, V., Wuersch, L., Islam, M. R. & Hicks, J. Indicators of community disaster fatigue: A case study in the new south wales blue mountains. *Int. J. Disaster Risk Reduction* **95**, 103831. https://doi.org/10.1016/j.ijdrr.2023.103831 (2023).
23. Shah, S. A., Seker, D. Z., Hameed, S. & Draheim, D. The rising role of big data analytics and iot in disaster management: Recent advances, taxonomy and prospects. *IEEE Access* **7**, 54595–54614. https://doi.org/10.1109/ACCESS.2019.2913340 (2019).
24. Nussbaumer, A., Pope, A. & Neville, K. A framework for applying ethics-by-design to decision support systems for emergency management. *Inf. Syst. J.* **33**(1), 34–55. https://doi.org/10.1111/isj.12350 (2023).
25. Koopman, P. Wagner, Michael: Challenges in autonomous vehicle testing and validation. *SAE Int. J. Transp. Saf.* **4**(1), 15–24. https://doi.org/10.4271/2016-01-0128 (2016).
26. Osoba, O.A., Welser IV, W., & Welser, W. An Intelligence in Our Image: The Risks of Bias and Errors in Artificial Intelligence. Rand Corporation, (2017)
27. Taeihagh, A. & Lim, H. S. M. Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks. *Transp. Rev.* **39**(1), 103–128. https://doi.org/10.1080/01441647.2018.1494640 (2019).
28. Matthias, A. The responsibility gap: Ascribing responsibility for the actions of learning automata. *Ethics Inf. Technol.* **6**, 175–183 (2004).
29. Leenes, R. & Lucivero, F. Laws on robots, laws by robots, laws in robots: Regulating robot behaviour by design. *Law Innov. Technol.* **6**(2), 193–220 (2014).
30. Butcher, J. & Beridze, I. What is the state of artificial intelligence governance globally?. *RUSI J.* **164**(5–6), 88–96. https://doi.org/10.1080/03071847.2019.1694260 (2019).
31. Kim, S. Crashed software: Assessing product liability for software defects in automated vehicles. *Duke L. & Tech. Rev.* **16**, 300 (2017).
32. Leenes, R. et al. Regulatory challenges of robotics: some guidelines for addressing legal and ethical issues. *Law Innov. Technol.* **9**(1), 1–44 (2017).
33. Tan, S. Y., Taeihagh, A. & Tripathi, A. Tensions and antagonistic interactions of risks and ethics of using robotics and autonomous systems in long-term care. *Technol. Forecast. Soc. Chang.* **167**, 120686. https://doi.org/10.1016/j.techfore.2021.120686 (2021).
34. Lele, A. *Disarmament* 217–229 (Arms Control and Arms Race. Springer, Singapore, 2019).
35. Roff, H. M. The strategic robot problem: Lethal autonomous weapons in war. *J. Military Ethics* **13**(3), 211–227. https://doi.org/10.1080/15027570.2014.975010 (2014).
36. Firlej, M. & Taeihagh, A. Regulating human control over autonomous systems. *Regul. Govern.* **15**(4), 1071–1091. https://doi.org/10.1111/rego.12344 (2021).
37. Scharre, P. *Autonomous weapons and operational risk* (Center for a New American Security Washington, DC, 2016).
38. Solovyeva, A., & Hynek, N. Going beyond the" killer robots" debate: Six dilemmas autonomous weapon systems raise. *Central Eur. J. Int. Security Stud.* **12**(3) (2018)
39. Papagiannidis, E., Enholm, I. M., Dremel, C., Mikalef, P. & Krogstie, J. Toward ai governance: Identifying best practices and potential barriers and outcomes. *Inf. Syst. Front.* **25**(1), 123–141. https://doi.org/10.1007/s10796-022-10251-y (2023).
40. Ghallab, M. Responsible ai: requirements and challenges. *AI Perspect.* **1**(1), 3. https://doi.org/10.1186/s42467-019-0003-z (2019).
41. Díaz-Rodríguez, N. et al. Connecting the dots in trustworthy artificial intelligence: From ai principles, ethics, and key requirements to responsible ai systems and regulation. *Inf. Fusion* **99**, 101896. https://doi.org/10.1016/j.inffus.2023.101896 (2023).
42. Khanna, S., Khanna, I., Srivastava, S. & Pandey, V. Ai governance framework for oncology: Ethical, legal, and practical considerations. *Q. J. Comput. Technol. Healthc.* **6**(8), 1–26 (2021).
43. Koshy, R. & Elango, S. Multimodal tweet classification in disaster response systems using transformer-based bidirectional attention model. *Neural Comput. Appl.* **35**(2), 1607–1627. https://doi.org/10.1007/s00521-022-07790-5 (2023).
44. He, K., Zhang, X., Ren, S., & Sun, J. Deep residual learning for image recognition. CoRR arXiv:1512.03385 (2015)
45. TensorFlow: Tokenizer - TensorFlow Keras Preprocessing. https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer. Accessed: 2024-08-12 (2024)
46. Pennington, J., Socher, R., & Manning, C. GloVe: Global Vectors for Word Representation. Accessed: August 2024 (2014). https://nlp.stanford.edu/data/glove.twitter.27B.zip.

47. Pennington, J., Socher, R., & Manning, C. GloVe: Global vectors for word representation. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (2014). https://doi.org/10.3115/v1/D14-1162 .
48. Contributors, T. TorchVision: PyTorch's computer vision library (2016). https://pytorch.org/vision/stable/index.html
49. Pedregosa, F. et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
50. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. Openai gym. CoRR arXiv:1606.01540 (2016)
51. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T.P., Harley, T., Silver, D., & Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. CoRR arXiv:1602.01783 (2016).

Author contributions

J.G.D conceptualized the study, developed the structured decision-making framework, conducted the analysis, and drafted the manuscript. A.Z provided guidance on the theoretical framework and critical revisions to the manuscript. N.R.G offered expertise in AI governance, provided feedback during the research process and on the manuscript. M.A.C contributed to the methodological design, interpretation of results and contributed to substantive revisions to enhance the manuscript's intellectual rigor, and led the supervision of the project. All authors reviewed and approved the final manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to J.G.D. or A.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.