Project Report

AI-Driven Disaster Response Triage
(Sumida Ward Deployment)

AYADEN Lina
Master 1 I2D

January 2026

1. The Problem and Context

For my project, I chose to model a disaster response scenario in Tokyo, focusing on the immediate aftermath of a Magnitude 7.3 earthquake in Sumida Ward. The main issue I wanted to tackle is the problem of ”information saturation” that happens right after a disaster. In this scenario, the Honjo Fire Station Command Center would be flooded with hundreds of simultaneous alerts from civilian calls, messages, and sensors. This creates a dangerous bottleneck where dispatchers are overwhelmed and might accidentally send rescue teams to minor incidents while missing critical structural failures nearby.

To solve this, I developed a triage algorithm that ranks these incoming signals within a 5 kilometer radius of the fire station. Sumida Ward is a unique challenge because it is a patchwork of two different eras.

- **The Modern Era:** It houses the Tokyo Skytree, one of the safest, most engineered structures in the world.
- **The Post-War Era:** Just blocks away, there are dense neighborhoods of wooden housing from the 1950s.

Because the terrain is so varied, a report of fire means something completely different depending on where it happens. To handle this, I partitioned the area into five zones.

1.1 Zone Risk Levels

I prioritized incidents based on where they happen. I divided the area into five zones, assigning each a Base Risk Score.

- **Zone 1: Kyojima (9.0).** This is the highest priority. It is a dense area of old wooden housing with narrow streets. The danger here is rapid spread. A single fire threatens to burn down the entire neighborhood because trucks struggle to access these roads.
- **Zone 2: Kinshicho (7.0).** A busy commercial hub. The risk comes from population density and threats like falling glass or traffic jams.
- **Zone 3: Skytree (5.0).** The tower is very safe. The risk is mostly internal, like people trapped in elevators or panic in the shopping complex.
- **Zone 4: Kameido (4.0).** Concrete apartment blocks. These are relatively safe compared to wooden houses, so ranking depends mostly on the message content.
- **Zone 5: Park (2.0).** Open ground and structurally very safe, but carries a Tsunami risk due to the river.

2. Defining the Criteria

I built the decision model around six core criteria (g_1 to g_6) plus a special Veto trigger. I used a "Points System" (Total = 39 points) to weigh them.

2.1 Main High-Impact Factors

- **The Veto Trigger (g_7) - 10 Points.** This is the highest weight. It acts as an override for "Kill Words" like Tsunami. It uses a Veto Threshold ($v = 0.9$) to ensure critical alerts are never outranked.
- **Severity (g_1) - 9 Points.** This measures the direct physical threat. Saving lives is the primary goal.
- **Zone Vulnerability (g_4) - 8 Points.** This accounts for strategic risk. Alerts from Kyojima get high priority because a small fire there can become a catastrophe.

2.2 Logistical and Secondary Factors

- **Proximity (g_3) - 5 Points.** Prefers incidents closer to the station so units can respond faster and save more people per hour.
- **Battery Life (g_5) - 3 Points.** A tie-breaker to find victims before their phone dies and we lose their GPS signal.
- **Panic (g_2) - 2 Points.** While panic can be noise, it is also a risk. A panicked crowd in a dense area can lead to stampedes or people blocking exits. I included this to add urgency to situations that might spiral out of control.
- **Accessibility (g_6) - 2 Points.** Our heavy fire trucks need clear roads. If a path is reported as blocked or collapsed, the system recognizes that reaching this spot will be difficult and lowers the priority slightly compared to reachable locations.

3. How Data is Processed

The system translates raw data into numbers. Human texts go through an NLP engine, while IoT sensors are trusted immediately. I only considered the first 25 seconds after the quake to take a "snapshot" of the situation.

- **g1 Severity:** I used a pre-trained model to classify text into labels like Life Threat or Collapse.
- **g2 Panic Level:** I look for Caps Lock shouting, exclamation marks, and emojis. High panic suggests a crowd might be losing control.
- **g3 Proximity:** Calculated using GPS coordinates to determine distance from the Honjo Fire Station.
- **g4 Zone Risk:** A lookup assigns risk based on the neighborhood (e.g., Wooden vs. Concrete).

- **g5 Battery Life:** Extracted from metadata. Lower battery increases priority to find people before they go ”dark.”
- **g6 Accessibility:** The system scans for keywords like blocked, rubble, or collapsed. If these are found, the score reflects the logistical difficulty of the rescue.
- **g7 Veto Trigger:** A safety override for massive threats like Tsunami or Gas Leaks.

4. The Ranking System (ELECTRE III)

I used the ELECTRE III method because it is non-compensatory. In a simple sum, a good score could hide a disaster. ELECTRE III compares alerts against each other and uses veto thresholds to prevent this.

I standardise the data so ”more is better” for every criterion. I defined three thresholds for each:

- **Indifference (q):** The margin where two scores are considered equal.
- **Preference (p):** The gap required to say one alert is strictly better.
- **Veto (v):** The valve that stops a critical threat from being outranked.

The script calculates a **Net Flow Score**, which is the difference between the sum of alerts a specific alternative outranks and the sum of those that outrank it.

5. Results and Analysis

I limited the analysis to a strict 25-second snapshot after the earthquake to simulate the first critical ”Pulse Check” of the city. To interpret the results effectively, I organized the data in two ways:

5.1 The Logic of Zones and Ranking

- **The Logic of Buckets:** I grouped every alert by its zone (e.g., Kyojima, Park). This allows for logistical efficiency, as a commander can handle all nearby tasks in one deployment.
- **Global Ranking:** I calculated a Global Rank (1 to 50) based on the urgency score. This ensures that scarce resources always go to the absolute worst incidents in the city, regardless of where they are.

5.2 Language and NLP Implementation

In order to perform the analysis more easily and ensure the results are clear for the reader to understand, I used simulated data in English and processed it using English-language NLP libraries. It is important to note that the same concept would work for Japanese data as well. Implementing this in Japanese would require a few tweaks, specifically replacing the English model with a Japanese-specific transformer (like *cl-tohoku/bert-base-japanese*) and using a tokenizer like *MeCab* to handle word segmentation.

5.3 Results for Kyojima (High Risk)

Here are the results for the Kyojima sector, sorted by their Global Rank:

Global Rank	NetFlow	Status	Message Text
#1	33.70	EXTREME	ACTIVE_FIRE_REPORTED_CONBINI_FLAMES_VISIBLE
#3	30.39	EXTREME	Water line rupture sustained flooding spreading to sidewalk
#5	26.66	EXTREME	elderly person trapped upstairs stairs collapsed help
#14	21.40	EXTREME	strong gas odor possible leak near school
#15	20.18	EXTREME	strong gas odor possible leak near school
#33	-11.79	Standard	Traffic signals down vehicles colliding at intersection
#34	-13.77	Standard	Shaking slowing maybe but aftershocks still strong
#35	-16.32	Standard	under desk shaking hard cant type well... help

I partially disagree with the AI here. It placed the trapped person (Rank 5) above the gas odor (Rank 14). While I believe the gas is the bigger contagion threat in a wooden neighborhood, the AI prioritized a confirmed life threat over a potential one.

5.4 Results for Park (Low Risk)

Here are the results for the Park sector:

Global Rank	NetFlow	Status	Message Text
#16	20.08	EXTREME	Sumida River showing unusual wave motion near embankment Tsunami
#17	19.16	EXTREME	Major road split open near bridge traffic halted
#41	-21.24	Standard	Ground surface liquefied mud coming up through cracks in road
#46	-24.11	Standard	OH GOD EVERYTHING IS FALLING PLEASE MAKE IT STOP
#47	-25.68	Standard	AUTOMATED_ALERT FALL_DETECTED USER_NOT_RESPONDING
#48	-26.07	Standard	WHERE IS MY FATHER HE WAS JUST HERE PLEASE RESPOND
#50	-27.09	Standard	help help help battery low

The AI correctly found the Tsunami as the only real threat in this zone. It ranked vague calls lower without deleting them, allowing dispatchers to check on them later.

5.5 The Skytree Outlier

Rank #12: *"Tokyo Skytree visibly swaying from my location this is terrifying."* This is a False Positive. The tower is designed to sway. However, in emergency response, False Positives are better than False Negatives; I would rather the system scream for a swaying tower than stay silent about a trapped victim.

6. Conclusion

In summary, the results from the Kyojima and Park sectors show that the ELECTRE III model successfully balances immediate life threats with strategic neighborhood risks. I found similar consistent results when analyzing other zones and testing the model with larger datasets, which suggests that this triage logic is reliable and scalable.

Even with the occasional false positive, the system effectively filters through the massive influx of data without ignoring vague alerts that might later become important. I believe this solution is a very strong approach to managing information overload during the first critical minutes of a disaster.