Dual-Determinant Clustering Algorithm for Optimization of Disaster Response Logistics

.
.

Sampson Akwafuo
Department of Computer Science
California State University, Fullerton, United States
0000-0001-8255-4127

Asad Abdul
Department of Computer Science
California State University, Fullerton USA
0009-0002-9258-7170

***Abstract* — Prompt response to disasters and emergencies are essential in reducing their impacts. Emergency Operation Centers (EOCs) and Points of Dispensing (PODs) are vital components of emergency preparatory logistics. Efficient placement of these EOCs (depots) is extremely needed to ensure timely and result-oriented intervention during sudden epidemic outbreaks or rapidly changing disasters. Using data from disparate sources: population, environmental, geo-spatial and resource availability, a novel Dual-Determinant Clustering Algorithm (DDCA) is developed, using a modified K-Means clustering algorithm. The algorithm is able to simulate disaster scenarios, in radically changing situations. The algorithm offers an improved time complexity while ensuring stability and efficiency in the clustering output. In this paper, we develop a real-time algorithm and a decision support system for optimizing emergency response logistics. Application of the model to a real-life case study underscores its usefulness.**

Keywords—*Disaster Management, Disease Outbreaks, Emergency Response Logistics, Optimization Algorithms, Ebola Health Informatics*

I. Introduction

In recent years, the frequency of occurrence of disease outbreaks and high-impact disasters have continued to increase. This underscores the importance of more elaborate and concerted efforts toward preparing and implementing efficient emergency response logistics. This is even more important in Low-and Middle-Income Countries (LMICs), where relevant infrastructure for predicting and monitoring disaster and disease outbreaks may not be adequate. The tragic floods that swept Southeast Morocco or the Ebola epidemic in West Africa demonstrate what disastrous effects logistical failures can lead to and how much such incidents are in dire need of quick and efficient emergency intervention. The central issue in this difficulty is the issue of positioning in supply depots and allotment of resources, a complicated optimization issue commonly known as the depot location-allocation issue. The main objective is to identify optimal locations for Emergency Operation Centers (EOCs) and Points of Dispensing (PODs) to efficiently accommodate a dynamic population working under limited conditions. The conventional methods of solving this issue have been relying highly on static facility location models and primitive clustering algorithms such as the K-Means[1]. Although these models have availed themselves useful in commercial logistics and in a stable environment, their weakness is evident in times of emergency. Their assumptions are based on fixed locations of demand, stable infrastructure, and stable resources, which are not realistic in real-life crisis circumstances. Route planning is one of the most difficult and fundamental problems in emergency logistics management [3].

In view of these limitations, a machine learning-based, adaptive and context-dependent algorithm is proposed: he Dual-Determinant Clustering Algorithm (DDCA). It uses strategies that broaden the approaches to the traditional clustering to incorporate two essential determinants namely the geographic accessibility and resource demand capacity. Compared to traditional clustering methodologies which only concern themselves with the minimization of intra cluster distances, DDCA seeks to provide geographically and logistically tenable clusters. Each cluster is planned to be within the reach of one of the depots with the relative urgency, as well as the relative difficulty of access of each region, being taken into consideration [2].

The DDCA framework beneficial in dynamic disaster condition where the actual situation in real-time, route interruptions, and population needs are not known. It uses a variety of heterogeneously collected data inputs, such as population density, the availability of a road network, the state of the environment, and the stockpiles of medical resources, to produce clusters that are not merely optimal during computation. It improve son previous clustering approach that utilizes capacity constraints [3]. Moreover, the algorithm is designed such that it is efficient when it comes to using time, which means that the re-computation process could be carried out relatively fast once new information is present. This enables the emergency managers to keep adjusting their response plans on a continuous basis as opposed to having mechanisms that are built on rigid models and would become out of date within hours.

The system has the potential to greatly improve disaster response management in low- and middle-income countries (LMICs), where the infrastructure is significantly weak, and the resources are scarce. It also applies to high-income regions, where a single instance of wide-scale disaster, say, a hurricane or a wildfire, or a pandemic, is likely to disrupt even the sturdiest logistics systems. Resource allocation is a universal requirement in emergency management in terms of quickness, fairness and strategy. The framework provided by DDCA does not only satisfy the technical requirements of this urgency, but also goes along the lines of humanitarian requirements of equity, speed and efficacy.

A. Problem Definition

Disasters strike with little warning, leaving communities desperate for aid. The aftermath of events like devastating floods in Morocco and epidemics in West Africa underscores an urgent global problem: getting life-saving supplies to people in need, fast and efficiently. In Morocco, heavy rains have triggered flash floods that washed out roads and isolated entire villages, making it nearly impossible for relief trucks to reach those cut off. During West Africa's Ebola epidemic, already fragile infrastructure was further crippled at one point over a million people were indirectly affected as regular supply routes were disrupted by the outbreak's spread. In that crisis, the World Food Programmed had to mount a massive logistics operation, moving huge volumes of medical supplies and staff in a race against time. These real-world examples highlight what's at stake: when disaster logistics falter, lives are on the line. Conversely, when emergency logistics are well-planned and executed, they dramatically reduce the human cost of catastrophes. In practical terms, responders face multiple intertwined challenges in post-disaster logistics, which current methods address only partially:

Depot Location-Allocation: Determining how many supply depots are needed and where to put them is exceedingly difficult in a disaster. Planners must balance coverage of all affected areas against travel times and stock availability, all while guessing how the situation will evolve. A poor choice can mean some communities are too far from help or supplies sit idle in the wrong place. Traditional location models don't easily accommodate the rapid changes in "demand points" that occur in, say, a fast-changing outbreak or a flood that progressively affects new regions. The challenge is to place depots optimally so that even as roads reopen or new areas are hit, aid is never too far away.

Resource Optimization: Resources in disasters are limited there are only so many relief kits, medicines, or rescue teams available, especially in low-resource settings. The key question is how to allocate these scarce resources effectively across depots and response teams. This involves deciding how much of each supply should go to each depot and which area gets priority when not everyone can be served at once. It's a complex optimization under uncertainty: sending too many resources to one region might leave another critically undersupplied. Yet holding back supplies "just in case" can cost lives in the immediate term. Classical models like the capacitated facility location can handle fixed capacities, but they rarely incorporate priority of needs or the option to re-distribute on the fly. The optimization must also consider that different items (food, water, medicine) have different importance and that needs can change rapidly (for instance, an area could suddenly require more water if local supply becomes contaminated). Achieving the least-cost, highest-impact use of resources is vital – in fact, using the least time and distance to deliver aid is most critical in low-resource regions where every fuel drop and hour of labor counts.

Real-Time Routing: Once depots and supplies are in place, the next challenge is delivering aid to countless scattered drop-off points under chaotic conditions. Routes must be planned for trucks or helicopters to get supplies from the depots to villages, shelters, or field hospitals. In a stable scenario, this is the classic Vehicle Routing Problem (VRP) – already complex – but here it must be solved in real time and often with incomplete data. Roads may be blocked by debris or floodwaters, and new information comes in continuously (e.g. a bridge collapse). Emergency vehicles might also have different capacities or capabilities, from large trucks to small boats or off-road vehicles, adding another layer of complexity. Routing in a disaster is a moving target: a path that was optimal an hour ago may become impossible or suboptimal after an aftershock or a sudden change in patient evacuation priorities. Current routing algorithms, even advanced ones, typically assume a static network and demand set while computing routes to a destination our DDCA includes a sub- algorithm for managing sudden situations in real-time, as described in section 3.

II. LITERATURE REVIEW

Machine learning and artificial intelligence algorithms can greatly improve disaster and pandemic management systems. By using these algorithms, authorities will be able to predict the emergence of any danger, forecast the trend of future outbreaks, explain what materials will be required and present the most effective evacuation strategies. Employment of K-means clustering can enable one to organize various regions, calculate logistics and resource management without complications. These types of models help manage transportation networks that must be flexible moment by moment. Critical tasks, for instance, examining satellite photographs, supervising flooding, spotting blockages on roads or telling if buildings are harmed are made easier by advanced architectures like CNNs and LSTMs. Having these tools saves time by quickly taking data received and turning it into useful information for rapid response[5]. Such systems are put into use during every step of disaster management from risk forecasting to reacting to the situation and organizing recovery. As an area recovers, regression models predict the supply need, the required number of workers and the needed infrastructure, making planning for the future easier.

Some experts use clustering to assemble areas that have the same or close risk levels in flood preparedness. With these methods, the positions of aid centers can be identified most efficiently. The systems address historical flood severity, size and type of potential barriers and reachability of the affected area. By locating relief centers properly, emergency help reaches everyone in the disaster zone more quickly. Clustering helps ensure all areas get equal resources and delivery drivers use the shortest possible routes [6]. Adopting this technique cuts time and makes the rescue teams less busy, so they can respond better to urgent incidents. The most urgent shipments are put in groups, so they are handled faster and make a bigger difference in relief. Using real-time statistics makes logistics networks more flexible and makes sure help gets to the areas in the greatest need first. Tests with such methods have proved that there are improvements in both the extent of a campaign and its performance. Clustering in logistics networks is important because accuracy and speed are essential. Goods are organized and assigned to clusters by considering how urgently they are needed, their demand pattern and past use. For example, the way medical supplies, food and water are sent will vary depending on how the recipient cluster is affected by the crisis. Rush deliveries use fast express shipping, but big deliveries prioritize saving fuel[7]. Delivery windows are set, and dispatch schedules are changed automatically thanks to the use of algorithms.

Clustering is part of the multi-agent approach applied in search and rescue. Robots, drones and autonomous vehicles are given to teams who must work within certain clusters. The boundaries of these clusters are set using terrain difficulty, the quality of reception and the number of potential victims. Within the clusters, the agents collaborate closely and swap information with others. The use of ML helps them make sure that no topic is missed and that nothing is repeated. If one drone notices a fallen structure in its cluster, it sends out the info to nearby agents who can go give aid or share the news with those on the ground. As a result, clustering provides both a location-based and an operational foundation for the plan[8], [9].

Dual-Determinant Geo-Clustering Algorithm

Input: *K = number of potential clusters;*
L = {l_1, l_2, ..., l_n}: List of locations to be clustered;
D = {d_1, d_2,}: Two initial cluster centroids
Output: *C = {c_1, c_2, ...c_k):* Cluster centroids
Ei: { E(i)|i = 1, 2, ... n} *: A list of clusters*

$N_k = 2$
foreach d_1, d_2, *do*
  calculate $argDist_1, argDist_2 \quad \forall\, d_1, d_2\ \varepsilon\ D$
  if $argDist_1$, < $argDist_2$, ***then***
    d_1( )← l_i ▶ *assign l_i to cluster d_1*
  else d_2( )← l_i ▶ *assign l_i to cluster d_1*
  end if
end for
foreach $l_i \in L$, *do (i = 3, ...n)*
  calculate $argDist_{i1}, argDist_{i2}\ \forall\, i, j\ \varepsilon\ C$
  ▶*Haversine distances from centroids*
  c*all* ***heapSort***(d_i, n)
  N_k+= 2
  $(dist_{li})$ = largest ▶ *maximum distance from heapSort*
  l_i ← new cluster centroid
  if $argDist_{i2}$, < $argDist_{j2}$, ***then***
  d_{i1}( )← l_i ▶ *assign the location to cluster d_{i1}*
  else d_{i2}( )← l_i ▶ *assign the location to cluster d_{i2}*
  end if

Fig. 1: Geo-clustering algorithm for each cluster

One of the most important innovations is the addition of obstacle-sensitive clustering algorithms, which differentiate the formations of clusters to identify damaged or unsafe roads, blocked bridges or any other areas that are not accessible. Clusters promoted through such techniques are not only spatially sensible but also practically accessible which is a crucial factor in disaster response. Obstacle or constraint-aware clustering has been found to minimize the probability of the infeasibility or safety of route plans and has therefore been demonstrated to increase efficiency and the resilience of humanitarian operations in a variety of studies. Dynamic streaming data, e.g., live weather feeds or changing flows of the population assist in building these predictive machine learnings models[10].

III. METHODOLOGY

The DDCA uses various technologies to achieve a real-time optimization of clustering operations and emergency response management in the target region. As for our website we have utilized Stream lit as our baseline for testing and development. This allowed for the team to quickly visualize and debug clustering results before migrating to a more robust full-stack implementation. Understanding the various circumstances that contribute to our Depot Location Allocator allows us to better prepare for disaster. Our utilization of various visualization libraries such as leaf map and folium have allowed us to render our data points and cities on a geographical map to better understand the distribution of emergency centers across a given region. These tools enable us to project real world coordinates offering valuable insights on spatial regions and more importantly how to respond to these in an orderly manner. Such visualization supports better planning and faster decision-making in dire situations.

A key component of our methodology lies in the integration of various **machine learning techniques**, specifically **clustering methods**. To properly group and assign our data valuable locations regarding their emergency operational centers clustering was required[11]. Our specific usage of the **K-Means clustering algorithm**, from the Sci Kit Learn library, has permitted us to accurately cluster together locations that will report to one emergency center in an effective manner maximizing the usage of each center per region. K-Means will provide an efficient spread of the workload among the depots since it will group the location based on spatial distance. The other clustering algorithm that we applied was the DBSCAN clustering, Sci Kit Learn, which has enabled us to compare the results of the different techniques of clustering with one another and present the optimal output of the many reachable locations per emergency operation center. The characterization of density between cluster and outliers using DBSCAN was helpful in measuring the robustness of K-Means on more irregular datasets[12]. The full-stack program of the web application enables its user to communicate with the user interface (UI) that is designed to be simple, but efficient and is being given in EJS with the presentation in stylish modern technologies such as CSS. It had an easily understandable navigation bar, card-written content and responsiveness with hover and transition functionalities to enhance the user experience. Subsequent versions utilize React.js which makes the system more dynamic by loading contents dynamically. This facilitated ease in scaling of interface and designing on multiple devices. Details of the algorithm, can be seen in Fig. 1

Procedure for Sorting Locations in a Cluster

procedure HeapSort (n[], i)
$n = L$ = number of locations
size ← 0;
for $i = n/2$ ***downto*** 0
  swap $n(1)$ with $n(i)$
  $n.heapSize = n.heapSize - 1$
end for
Max-Heapify (L, 1)
l: = 2i
r: = 2i+1
if $l \leq n.heapSize$ and $(n[l] > n[i]$
  $largest = l$
else $largest = i$
end if
if $r \leq n.heapSize$ and $(n[r] > n[largest]$
  $largest = l$
else $largest = r$
end if
if $largest \neq i$
  swap $n(i)$ with $n(largest)$
  Max-Heapify (n[],largest)
end if

Fig. 2: *HeapSort* procedure for finding largest *argDist*

The backend structure of the application was constructed with Node.js and Express.js that facilitate the safety of data transfer and high-performance operations on the server. It is controller logic to be added to the system and its authentication system having support of hashed password

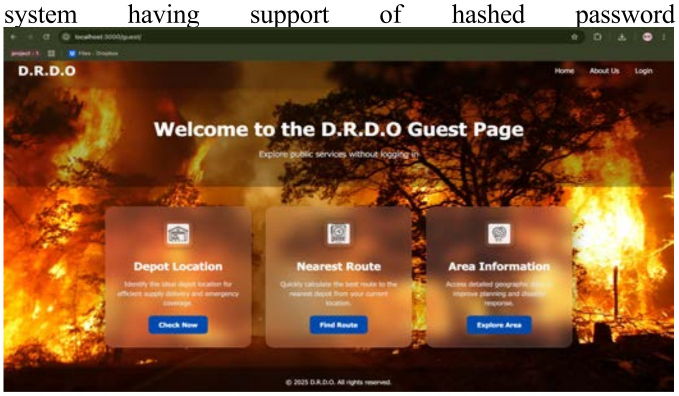

Figure 3: Components of the higher-level menu

saving, JWT-based software session, and multi-factor authentication through email verification. The Mail trap was employed on test email handling at the stage of development so that production ready email delivery could be faked to have no real users being impacted.

A. *Maintaining the Integrity of the Specifications*

All system and user states are kept under control with the help of MongoDB, which guarantees effective storage and retrieval of the user-related documents, user profiles, cluster tokens, etc. There is also the integration of Axios in the handling of user API requests between and among the client and the server, providing an ideal real-time experience that does not call for reloads of the page. The responsive nature of the project has been met with the philosophy of responsive design through flexible grid layout and the use of CSS media queries where the platform will be a hundred percent usable on desktop and tablets, as well as mobile. A combination of **HTML5, CSS3, and JavaScript** connects the frontend styling and layout control, delivering both form and functionality alike.

Our deployment type has evolved over time through multiple iterations, ultimately it was decided that **Amazon Web Services (AWS)** was most suitable for hosting our web service. The scalability, flexibility, and control of AWS offers support to a complex, full-stack system like ours, which integrates backend services, machine learning models, and a NoSQL database while allowing for the expansion of features later. Initially we considered deploying the service through **Git hub Pages**. However, when hosting with this framework it is limited to static content such as portfolios and documentation. These limitations have made it incompatible with our technology stack, like Node.js, Python-based clustering algorithms, and ".

IV. RESULT AND DISCUSSION

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

A. *Website Overview*

This web application provides a seamless and interactive user experience designed with both functionality and aesthetics in mind. It is constructed according to the current best practices and is designed around usability through navigability, visual readability and functionality on all devices. The site architecture will be built based on a set of effective frontend and backend technologies, which guarantees the scalability and the absence of any problems during its work.

B. *Frontend Design*

It has a streamlined attractive layout, as the user interface (UI) is simple and beautiful. Responsive, mobile first approach is at the heart of the design as it will assure that the given application is working perfectly on devices of different screen size: both desktops and mobile devices. The site uses custom

CSS styling and offers a clean and uniform appearance on every page.

Navbar: The navbar is drafted in such a way that the process of user interaction is intuitive and has bright ways to all the necessary pages. It has links to important pages including home page, logging page, guest access page among others. Sleek, semi-transparent effect of the navbar is done with backdrop-filter to add to the visual aesthetics provided in a way that does not interfere with functionality.

Card Layouts: Within the home page, card layouts are one of the most visible elements, which are present to render different sections and options which are very much readable and arranged in a visual form. Every card also contains hover effects where necessary to tempt users to be involved in significant sections and ensuring that the interface becomes lively and satisfied with touch-to-touch involvement.

Interactive Elements: The webpage includes buttons and clickable elements designed for smooth transitions.

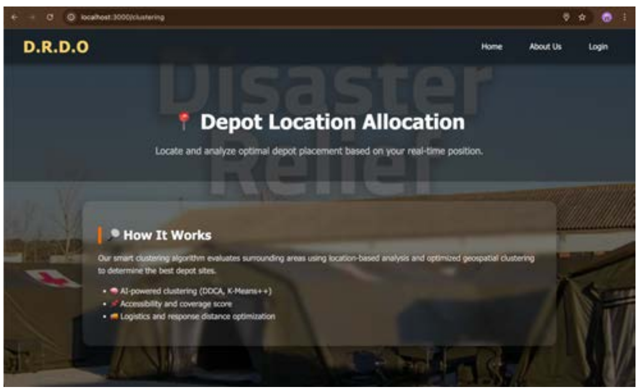

Fig 4: Depot Location Allocation Module

C. *Functionality*

The application leverages a robust backend framework, enabling users to interact with the site smoothly, without experiencing significant delays or performance issues. Below are the core functionalities that ensure a seamless experience for all users:

User Authentication: Secure login and registration processes are in place to ensure that only authorized users can access specific features of the site. The authentication system includes **multi-factor authentication (MFA)** for added security, coupled with **email verification** to ensure genuine user registration.

Dynamic Content: The application dynamically loads content depending on user input. For example, users can explore a range of different content pages without requiring a page reload, thanks to the seamless integration of **React.js**.

D. *Responsive Design*

The site is fully **responsive**, adapting its layout to various screen sizes, ensuring that it is user-friendly on mobile devices, tablets, and desktops. Media queries and flexible grid layouts allow the content to adjust to different screen resolutions, ensuring an optimal viewing experience no matter the device being used.

E. *Prototyping*

During the creation of our prototype, we intended on making the UI as simple as possible while still producing all the necessary working components for our program. As displayed in the figures the prototype walks through all the steps of what to do and how to input into our clustering algorithm with simple buttons. Throughout the prototyping process we edged out the difference between the components we wanted to keep and remove. Many of the main backend components have remained the same in the migration from the prototype to the final version such as elements like clustering, database retrieval, priority listing, etc. We then made the decision to elevate the UI/UX and create our final product with a responsive design that ensured an optimal user experience.

F. *Case Study*

To evaluate the real-world applicability of our proposed web service we conducted a case study which utilized multiple different test files including different formats such as CSV, Geo JSON, and Shapefiles. Our initial study began with a list of U.S. cities in a csv file format. The U.S cities CSV contained geographical coordinates (longitude and latitude), and this dataset served as the baseline to identify potential points of interest which in our case would be Emergency Operation Centers (EOCS). The coordinates were extracted from the CSV file and were loaded into a panda Data Frame which served as input coordinates to our clustering algorithm. The preliminary data cleaning required us to handle nullified values and verify that the data was clean such as ensuring accuracy of coordinates.

The coordinate data was passed through our DDCA model, which builds upon the traditional K-Means algorithm. In addition to the locational proximity, the algorithm accounted for another key factor to cluster distribution being the population density, which was a second determinant for the distribution. This approach allowed for our results in the clustering to be geographically logical and balanced especially in areas of high traffic/density. This made the DDCA more adaptable and efficient than the natural K-Means as with these heuristics the preparation for disaster scenarios is significantly increased. Not only does the service provide a quicker manner of preparation but an accurate provision and distribution of resources where in scenarios depots that require more support are appropriately accommodated. The comparison to the DBSCAN clustering algorithm for the same dataset was useful in identifying points of non-uniform density areas. However, it lacked a sense of structured optimization for strategic planning of these specific hubs of service. The DDCA provided a balance of computational efficiency and logistical relevance[13], [14].

The mapping libraries, folium and leaf map, were utilized to display geospatial maps appropriately with the conditions and markers applied to properly understand cluster center spreads and the points that respond respectively. Each cluster was drawn with a path to their cluster center to display which EOC would respond to the location region. The visual output confirmed that the algorithm was working properly to distribute clusters efficiently, minimizing the travel amongst the regions and their centers thus outputting the optimal regional coverage.

V. CONCLUSION

The means to implement effective emergency logistics in a timely and efficient manner are more vital than ever, given the increase in the frequency of natural disasters and the growing number of public health crises. Ranging from floods and earthquakes to epidemics and infrastructure collapse, disaster situations create a complex of issues to be addressed volatile demand and broken transportation links, scarcity of resources, the demand of fair allocation of relief. Although the traditional logistics and clustering models have been found to be effective in either a static or commercial scenario, there are often inferiorly placed to address such fluid-and-high-stakes environments. These models have the tendency to be inadequate when it comes to confronting reality, facing the limits of capacity, accessibility, urgency, and quick adaptability.

This paper offers the Dual-Determinant Clustering Algorithm (DDCA) as an in-depth and prospective answer to these quandaries. By incorporating not only geographic proximity but also capacity-based demand factors into the clustering process, DDCA offers a solid framework for optimal placement and allocation of Emergency Operation Centers (EOCs) and medical countermeasures, ensuring effective delivery of relief supplies. DDCA unlike the traditional algorithms, such as standard K-Means that primarily focus on the minimization of distance considered the logistical feasibility of each cluster in a way that does not congest any of the depots and focuses on metering high demands accordingly[15], [16]. This renders it especially applicable in active, limited resources operations where human lives rely on the speed and correctness of logistical decision-making. It could also handle fluctuations in input data and recalculate when the situation changed Such functionalities are critical in a real-life disaster situation where new information (e.g., blocked roadways, new hotspots, movement of people) constantly changes the environment in which one must work. Moreover, its balanced method has also been justified by the comparison to other clustering algorithms, such as DBSCAN, which proved beneficial to irregular data distributions but is missing the strategic value appealing to structured logistic planning purposes.

In the future, the study will provide new opportunities in the development of intelligent systems of disaster response. Other potential improvements on DDCA in the future are the additions of predictive analytics on anticipatory depot location, multimodal route optimization due to the use of different means of transportation, as well as reinforcement learning that can be updated over time based on the input of operational performance. Besides, social vulnerability measures and indicators of community resilience can also help situate DDCA.

The Dual-Determinant Clustering Algorithm, therefore, is a considerable step towards the optimization of post-disaster logistics. DDCA manages to fill this gap between computational modeling and real-time emergency response by offering algorithmic rigor with practical application and useful interface to the user. It allows emergency managers to make prompt, reasonable, and fair decisions as well as saves lives and allegedly minimizes suffering and builds the resilience of crisis-stricken communities.

VI. REFERENCES

[1] J. José, P. Medina, and R. B. Serrato, "Determination of delivery routes using the k-means and Elbow algorithms," 2021. [Online]. Available: www.jmess.org

[2] L. Tanti, S. Efendi, M. S. Lydia, and H. Mawengkang, "A Decision-Making Model to Determine Dynamic Facility Locations for a Disaster Logistic Planning Problem Using Deep Learning," Algorithms, vol. 16, no. 10, Oct. 2023, doi: 10.3390/a16100468.

[3] S. E. Akwafuo, A. R. Mikler, and C. Ihinegbu, "Geo-clustering model for optimizing locations of public health emergency operations and covid-19 vaccine distribution centers," in 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), IEEE, 2022, pp. 1108–1113.

[4] S. Akwafuo, A. R. Mikler, and F. A. Irany, "Optimization Models for Emergency Response and Post-Disaster Delivery Logistics: a Review of Current Approaches," International Journal of Engineering Technologies and Management Research, vol. 7, no. 8, pp. 35–49, 2020, doi: 10.29121/ijetmr.v7.i8.2020.738.

[5] L. Tanti, S. Efendi, M. S. Lydia, and H. Mawengkang, "A Decision-Making Model to Determine Dynamic Facility Locations for a Disaster Logistic Planning Problem Using Deep Learning," Algorithms, vol. 16, no. 10, Oct. 2023, doi: 10.3390/a16100468.

[6] T. D. C. Le, D. D. Nguyen, J. Oláh, and M. Pakurár, "Clustering algorithm for a vehicle routing problem with time windows," Transport, vol. 37, no. 1, pp. 17–27, May 2022, doi: 10.3846/transport.2022.16850.

[7] J. B. Sheu, "An emergency logistics distribution approach for quick response to urgent relief demand in disasters," Transp Res E Logist Transp Rev, vol. 43, no. 6, pp. 687–709, 2007, doi: 10.1016/j.tre.2006.04.004.

[8] V. Linardos, M. Drakaki, P. Tzionas, and Y. L. Karnavas, "Machine Learning in Disaster Management: Recent Developments in Methods and Applications," Jun. 01, 2022, MDPI. doi: 10.3390/make4020020

[9] M. Amin Amani, S. Asumadu Sarkodie, J. B. Sheu, M. Mahdi Nasiri, and R. Tavakkoli-Moghaddam, "A data-driven hybrid scenario-based robust optimization method for relief logistics network design," Transp Res E Logist Transp Rev, vol. 194, Feb. 2025, doi: 10.1016/j.tre.2024.103931.

[10] C. Liu, J. Yu, S. Han, and S. Liang, "Research on the Optimization of Emergency Material Distribution Route in 'Vehicle' Mode Based on K-Means Clustering and LK Algorithm", doi: 10.23977/jwsa.2022.040103.

[11] V. Chamola, V. Hassija, S. Gupta, A. Goyal, M. Guizani, and B. Sikdar, "Disaster and Pandemic Management Using Machine Learning: A Survey."

[12] Y. Li, H. Sun, and X. Liu, "Deep learning for rapid damage assessment in natural disasters," Remote Sens (Basel), vol. 10, no. 10, p. 1523, 2018.

[13] X. Xu et al., "PERMS: An efficient rescue route planning system in disasters," Appl Soft Comput, vol. 111, Nov. 2021, doi: 10.1016/j.asoc.2021.107667.

[14] J. Vargas-Florez, R. Medina-Rodríguez, and R. Alva-Cabrera, "A clustering optimization approach for disaster relief delivery: A case study in Lima-PerúPer´Perú."

[15] T. Wu, L. Huang, Z. Liang, X. Zhang, and C. Zhang, "A supervised learning-driven heuristic for solving the facility location and production planning problem," Eur J Oper Res, vol. 301, no. 2, pp. 785–796, Sep. 2022, doi: 10.1016/j.ejor.2021.11.020.

[16] X. Chen, L. Zhang, X. Lin, K. A. Skalomenos, and Z. Chen, "A clustering-based analysis method for simulating seismic damage of buildings in large cities," Eng Struct, vol. 307, May 2024, doi: 10.1016/j.engstruct.2024.11786