A Distributed Framework of Autonomous Drones for Planning and Execution of Relief Operations during Flood Situations

Zobia Zafar[1], Muhammad Awais[1], Abdul Jaleel[1], and Fiaz Majeed[2]
[1]Department of Computer Science, University of Engineering and Technology, Pakistan
[2]Department of Software Engineering, University of Gujrat, Pakistan

Abstract: *Every year, flood hits the world economy by billions of dollars, costs thousands of human and animal lives, destroys a vast area of land and crops, and displaces large populations from their homes. The flood affected require a time-critical help, and a delay may cause the loss of precious human lives. The ground rescue operations are difficult to carry out because of the unavailability of transport infrastructure. However, drones, Unmanned Vehicles, can easily navigate to the areas where road networks have been destroyed or become ineffective. The fleet participating in the rescue operation should have drones with different capabilities in order to make the efforts more successful. A majority of existing systems in the literature offered a centralized system for these drones. However, the performance of the existing system starts decreasing as the required number of tasks increases. This research is based on the hypothesis that a distributed intelligent method is more effective than the centralized technique for relief operations performed by multiple drones. The research aims to propose a distributed method that allows a fleet of drones with diverse capabilities to communicate and collaborate, so that the task completion rate of rescue operations could be increased. The proposed solution consists of three main modules: 1) communication and message transmission module that enables collaboration between drones, 2) realignment module that allows drones to negotiate and occupy the best position in the air to optimize the coverage area, 3) situation monitoring module that identifies the ground situation and acts accordingly. To validate the proposed solution, we have performed a simulation using AirSim simulator and compared the results with the centralized system. The proposed distributed method performed better than legacy systems. In the future, the work can be extended using reinforcement learning and other intelligent algorithms.*

Keywords: *Autonomous drones, flood relief operations, distributed systems, artificial intelligence, distributed collaboration.*

Received October 11, 2019; accepted July 14, 2020
https://doi.org/10.34028/iajit/18/1/3

1. Introduction

As a result of significant climate shift [4] and lack of dams [16], floods are the highest frequently occurring natural disaster events [1], causing significant losses to human life, livestock, and the property every year [2]. After the disaster hits, the first 72 hours are the most critical and life-threatening, which demands immediate and rapid rescue operations [19]. These responses require accurate and timely information [12], and that is only possible with the proper communication and management between victims, and various actors of response [3]. Owing to the collapse of physical infrastructures, legacy communication channels become ineffective [5] and ground movement almost become paralyzed.

Since the past few years, the community has experienced an extensive usage of drones to provide instant benefits to inhabitants and societies [13], especially in disaster situations [10, 12]. The tasks include collecting of information [11, 14]; planning for goods transportation [21] distributing of packages [15]; establishing of the network and communication [17, 22]; surveying and collecting of data [6, 9]; analyzing the situation and reducing of risk [18, 24]. Different situations demand different types of responses, such as the requirement of life-saving jackets, boats, and the supply of goods (food, medicine, etc.,). Therefore, to handle these situations, fleets of drones with diverse capabilities are required to take part in the rescue operations. For example, a drone can have a life-saving capability, while others can transport clothes, medicine, and food.

In order to optimize the relief operations, existing systems work mostly on cost minimization monitoring [13] and transport of goods [8, 15]. All these dimensions are essential; however, we have not found a work that automatically plans, executes and optimizes the rescue operations with a fleet of drones working under diverse capabilities to handle different types of situations on the ground. Therefore, a distributed framework of drones is required which enables the drones to collaborate, share information, and design the best strategy dynamically according to the situation, so that more lives can be saved.

The goal is to allow the drones with different capabilities to share and collaborate and form the best optimal strategy to complete maximum relief operations. More specifically, we have asked the following research questions.

- How can a communication mechanism be established within a network of drones in a disaster situation?
- How can a distributed ad-hoc network be created between different capabilities of drones so that they can collaborate to perform different relief operations?
- How can drones automatically reposition to cover a larger area under diverse capabilities?

The rest of the paper is divided as follows. Section 2 discusses the different solutions available in the literature that handles operations in a single and multi-drones' environment. In section 3, we proposed a distributed framework under the hypothesis that the fleet of drone work best with distributed coordination. The details of the experimental setup, experiments, and results are given in section 4. In section 5, we discuss the results with reference to our research questions. Finally, we conclude our findings in section 6.

2. Literature Survey

For a disastrous region, Chowdhury *et al*. [8] developed an integrated continuous approximation model to minimize the cost of allocating inventories and transporting emergency supplies using trucks and drones. The research concluded that the best possible distribution centers could reduce the overall cost for disaster operations. Erdelj *et al*. [11] concentrated on the combined features of the wireless sensor networks and multi-drones in the framework. This system provided effective results for information sharing, individual communication, and monitoring. For improving the delivery process, Rabta *et al*. [21] designed a model to minimize the total travelling distance of the drone under payload and energy constraints. The drones with fewer payloads performed well and completed the task in less time. Grippa *et al*. [15] proposed policy at the system level to control a fleet of drones for delivering goods from depots to customers. The policy stabilizes the system for all loads if the number of vehicles per depot is sufficient.

Ranjan *et al*. [22] presented sky help framework using drones to provide reliable communication infrastructure that increases the coverage area during an emergency scenario. The results showed that the packet error rate was significantly reduced from 0.98 to 0.0051. Kawamoto *et al*. [17] developed a resource allocation method based on Carrier Sense Multiple Accesses (CSMA) with collision avoidance protocol and time division to avoid conflicts when multiple UAVs transmit over WiFi infrastructure. The data was transmitted between drones and ground stations without packet conflicts, even when more than one drones transmit in the same frequency channel.

Dinesh *et al*. [9] developed a surveillance model using Unmanned Aerial Vehicles (UAV) for monitoring of the human body. The drones send live visual feedback to the monitoring stations and email alerts to the control stations. Choksi *et al*. [6] proposed cloud-based communication with drones using the Internet of Things (IoT) network for quick response strategy in case of disaster situations. This model collected and communicated the data generated by the sensors to the cloud platforms in real-time. Kumar *et al*. [18] designed a situation-aware conditional sensing algorithm for disaster-prone areas using drones with the help of IoT devices. The work improved the relief operations and communication among different agents.

Silva *et al*. [24] developed a procedure using UAV technology and geographic information systems that assisted stakeholders to form a distribution network during natural disaster response operations. This procedure helped the authorities for necessary aid supplies and reduced the time required for the decision making process. Table 1 shows a comparison of the proposed system with existing systems for relief operations in disaster-affected areas.

Table 1. Research matrix.

Research	Distributed Network	Sharing	Monitoring	Relief Operation	Classify Image through CNN	Optimal Positioning
Chowdhury *et al.* [8]	×	✓	×	×	×	×
Erdelj *et al.* [11]	✓	×	✓	×	×	×
Rabta *et al.* [21]	×	✓	×	×	×	×
Grippa *et al.* [15]	×	✓	×	×	×	×
Ranjan *et al.* [22]	✓	×	✓	×	×	×
Kawamoto *et al.* [17]	✓	×	×	×	×	×
Dinesh *et al.* [9]	×	×	✓	×	×	×
Choksi *et al.* [6]	×	✓	×	×	×	×
Kumar *et al.* [18]	×	✓	✓	×	×	×
Silva *et al.* [24]	✓	✓	×	×	×	×
Proposed Method	✓	✓	✓	✓	✓	✓

3. Proposed Solution

The proposed solution is developed for the fleet of drones taking part in the relief operation after the flood has hit an area, Figure 1. The suggested framework equips each member of the fleet with the different components, Figure 2, that enable drones to perform operations of communication, monitoring, network management, and realignment. Also, an additional controller component is proposed that allows inter-component communication. In the next sections, we explained all these components one by one.

Figure 1. A fleet of drones executing the rescue operation.

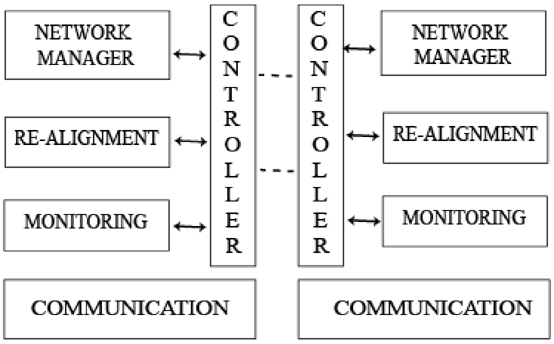

Figure 2. Components of the drone and communication model.

3.1. Communication Component

In the proposed system, three kinds of communication and collaboration occur within the drones: drone registration, situation alerts, and repositioning. When a drone enters into the existing surveillance area, it sends a broadcast message, Message Add Me Broad Cast, to all neighboring drones so they can add it into their Look Up table. If the neighboring drone adds a sender into its lookup table, it sends an acknowledgment message, Message Added Drone. A drone also sends the message, Message Situation, to its neighboring drones when it cannot handle the situation. Thirdly, a drone sends the message, Message Realignment, to another drone when its capability matches with that drone. The following are the details of these Foundation for Intelligent Physical Agents (FIPA) messages.

- **Message Add Me Broadcast** (inform: sender D: receiver {D1, D2, D3, ….}: content (Request for Addition received): language fipa-sl)
- **Description**: When a drone changes its location, it broadcast a message to drones existing in the network.
- **Message Added Drone** (inform: sender D: receiver {D1}: content (Added message received): language fipa-sl)
- **Description**: When the receiver drone D1 adds data of sender drones into its lookup table, it sends the acknowledgment message “Message Added Drone” to the controller that passes the message to the sender.
- **Message Alive** (inform: sender D: receiver (D1, D2, D3,….): content (=(Are You Alive)): language fipa-sl)
- **Description**: A drone Di sends a message, Message Alive, to verify whether another drone Dj, in its lookup table, is alive. If Dj sends back the acknowledgment, Di updates the timer for the drone in its lookup table.
- **Send Situation Message** (inform: sender D: receiver (D1, D2, D3, ….): content (Situation): language fipa-sl)
- **Description**: If a drone is unable to handle the situation, it sends message “Message Situation” to all drones in its lookup table.
- **Message Realignment** (inform: sender D: receiver (D1):content (Realign Position to avoid overlapping): language fipa-sl)
- **Description**: When drone Di identifies that another Drone Dj with the same capabilities is within its overage area, the Di sends message “Message Realignment” to Dj, so it realigns position to avoid overlapping.

3.2. Monitoring Component

Monitoring is the principal component of the proposed architecture that analyzes the whole situation and gets images of different places of affected areas. The trained Convolutional Neural Network (CNN), as given in [7], can be used to classify the images into different situations, Figure 3. In the first step, the convolution layer detects features of images; the feature map passes to the Rectified Linear Unit (RELU) layer that explores linearity. Next, the processed image forwards to the pooling layer that reduces the number of parameters. Then, the flattening process flattened the matrix into a vector and fed it into a fully connected layer. Finally, the system generates a situation that was occurring on the ground. The trained CNN was deployed on the drone's firmware, so it can process and categorize situations live.

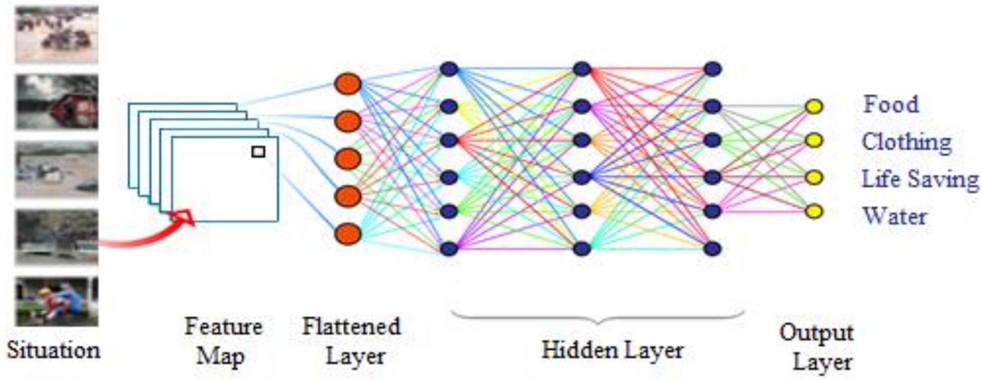

Figure 3. Convolution neural network for situation analysis.

3.3. Network Manager Component

This component manages the repository of the neighboring drones, and it is responsible for maintaining the network information. The primary functions of the component include the adding of a new drone in the lookup table, updating the lookup table after a specific time, and avoiding the drone overlapping with similar capabilities.

3.3.1. Add New Drone

When a message "Message Add Me Broad Cast" is received, the controller triggers the method "Add New Drone()", presented as Algorithm 1.

Algorithm 1: Add New Drone

Input: New-Drone

Output: Acknowledgement Message
if (New-Drone. Capability != This. Capability)then
New-Drone. expiry Time. set(100);
This. Drone Table .add(New-Drone);
Send Message(Drone. URI, Message Added Drone);
else Send Message(Drone.URI, Message Realignment);

- **Description**: If the capability of New Drone is different from its own capability, the controller adds the new drone into its Look Up Table. The controller also sets the expiry time of the drone to 100 seconds and sends the message "Message A dded Drone". In the case of the same capabilities, the controller sends the message" Message Realignment".

3.3.2. LookUp Table

The controller triggers after "Update Look UP Table()" and "Reset Expirt Time()" after every 10 seconds to update the lookup Table, presented as Algorithm 2A.

Algorithm 2A: Update Drone Table

Input :Drones In Look Up Table

Output: Send Message Or Deletion Of Drones

for each(D € Drones-in-Drone table)
if (D. expiry Time <10 && D.expiryTime > 0)then
Send Message(Drone. URI, Message Alive);
else if(Drone. expiry Time <= 0)
Look Up Table. Delete(D);
else D. expiry Time=D.expiryTime-10;

- **Description**: For each drone in Look Up Table, the algorithm checks the value of expiry time. If the expiry time of a drone is between 1 and 10, it sends "Message Alive" to verify whether the drone is still in its neighbors. In case, the expiry time is lesser or equal to 0, the algorithm deletes the drone from its Look Up Table. If the expiry time is greater than 10, the expiry time is decreased by ten every time.

3.3.3. Reset Expiry Time

Algorithm 2B: Update Drone Table

Input: Acknowledge-Drone

Output: Update Drone Table

Is Drone Exist=false;
for each (D € Drones-in-Drone table)
if (D == Acknowledge-Drone)then
D. expiry Time= 100;
Is Drone Exist=true;
else Add New Drone(Acknowledge-Drone)

- **Description**: When the live acknowledgment message is received, the controller calls the Algorithm 2B. The algorithm checks whether the sender is in the Lookup Table. If the sender is in it, the algorithm reset the drone expiry time; otherwise, it forwards the request to Add New Drone method.

3.3.4. Situation Handler

The controller triggers Algorithm 3 when it receives the "Send Situation Message."

Algorithm 3: Situtation Handle

Input: Situation
Output: Action or Send Situation Message
If(This. Capability == Situation)then
This. Handle(Situation);
else
for each(D € This. Drone Table)
Send Message(Drone.URI, Send Situation Message);

- **Description**: The method matches the situation with the capability of the current drone. If both match with each other, the current drone handles the situation. If both have different capabilities, the algorithm sends the message "Send Situation Message" to all other neighboring drones.

3.4. Realignment Component

This module optimizes the distribution of the drones with different capabilities in the disaster areas. For the best hovering position, the algorithm measured the area's radius covered by each drone using a grid-like structure. Each type of drone has a range of up to a fixed number of boxes. Drones with similar capabilities tend to remain separate from each other, whereas drones with different capabilities can overlap the boxes of others. In this way, drones with different capabilities are scattered in a specific arrangement in order to handle the situation efficiently. A sample of the grid structure is shown in Table 2. Here, F, C, W, and L represent Food, Clothing, Water, and Life Saving, respectively. An example of the location swapping to adjust drones position as per their capabilities is depicted in Figure 4, where D1 drone is swapping its position with drone D4 to handle the situation C1, according to Algorithm 4.

Table 2. Realignment grid.

<table>
<tr><td colspan="2">F</td><td>F</td><td>L</td><td>F</td><td>F</td><td colspan="3">F</td><td colspan="2">F</td><td>F</td><td>W</td><td colspan="3">F</td><td>F</td><td>.</td><td>F</td><td>F</td><td colspan="3">F</td></tr>
<tr><td colspan="2">F</td><td>C</td><td>L</td><td>C</td><td>C</td><td colspan="3">C</td><td colspan="2">C</td><td colspan="2">W</td><td>F</td><td colspan="2">L</td><td></td><td>.</td><td></td><td></td><td colspan="3">L</td></tr>
<tr><td colspan="2">F</td><td colspan="2">L</td><td>L</td><td>C</td><td>L</td><td>W</td><td>L</td><td>W</td><td>L</td><td colspan="2">W</td><td>F</td><td>W</td><td>L</td><td>W</td><td>.</td><td>W</td><td>L</td><td colspan="3"></td></tr>
<tr><td>F</td><td>W</td><td colspan="2">L</td><td></td><td>C</td><td colspan="3">W</td><td colspan="2"></td><td colspan="2">W</td><td>F</td><td colspan="2">L</td><td></td><td>.</td><td>L</td><td></td><td colspan="3"></td></tr>
<tr><td>F</td><td>W</td><td colspan="2">L</td><td></td><td>C</td><td colspan="3">W</td><td colspan="2"></td><td colspan="2"></td><td>F</td><td colspan="2">L</td><td></td><td>.</td><td></td><td></td><td colspan="3"></td></tr>
<tr><td colspan="2">W</td><td colspan="2">L</td><td></td><td>C</td><td colspan="3">W</td><td colspan="2"></td><td colspan="2"></td><td>F</td><td colspan="2">L</td><td></td><td>.</td><td></td><td></td><td colspan="3"></td></tr>
<tr><td colspan="2">W</td><td colspan="2"></td><td></td><td></td><td>W</td><td>C</td><td>L</td><td colspan="2"></td><td colspan="2"></td><td colspan="3">L</td><td>L</td><td>.</td><td>L</td><td>F</td><td>L</td><td>F</td><td>L</td></tr>
<tr><td colspan="2">W</td><td colspan="2">W</td><td>W</td><td>W</td><td colspan="3">C</td><td colspan="2">W</td><td colspan="2">W</td><td colspan="3">W</td><td>W</td><td>.</td><td>W</td><td>C</td><td colspan="3">C</td></tr>
</table>

Algorithm 4: Realignment Grid based Swapping

Input: Drone D
Output: Change Location List Location To Be Pushed
//List of position of drones Location To Be Pushed.
add(Location Of D From Left)
Location To Be Pushed. add(Location Of D From Right)
Location To Be Pushed. add(Location Of D From Top)
Location To Be Pushed. add(Location Of D From Bottom)
Var min Distance = 15 //Variable for min distance
Var min Location
//Variable for location of min distance For each (L€ Location To Be Push)
If (L. distance ¡min Distance)
min Distance=L. distance
min Location=L. Location
this. Move(min Location);

In realignment, as depicted in Figure 3, when D4 detects the situation C1, it checks whether the situation is within its capability, and if it is not, it passes it to the other drones. The drone D1 identifies that a situation related to its expertise has happened and require its services. If it D1 is free it sends its consent to the drone D4 to handle the situation. Thus, D4 let D1 (that has the capability C1) to respond to the situation. Then, both drones swap their locations.

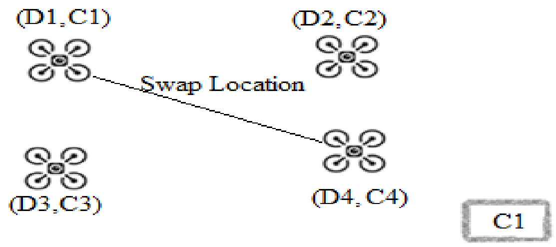

Figure 4. Drone adjust location with respect to capabilities.

3.5. Controller

The controller communicates with other drones and takes actions when necessary, according to Algorithm 5. The controller also triggers a number of actions at specific time intervals to keep the information updated. During the monitoring phase, each drone takes pictures of the affected area using cameras and passes these images to the trained convolution neural network, which categorizes the image and updates the controller on the situation. If the situation is within the drone's capabilities, the controller calls the situation handler. Otherwise, it broadcast the situation to all nearby drones in its lookup table. The lookup table is updated whenever a new drone comes near to any other drone. If both drones have the same capability, they can negotiate with each other about their optimal position.

Algorithm 5: Controller for Monitoring Phase

Input: Drone D, Messages
Output: Action
Thread. Timer(10)Send Request(Add New Drone);
Thread. Timer(10).Send Request(Update Drone Table)
Thread. Timer(10).Send Request(Realignment Drone)
if (Message. Type == Send Situation Message)
Thread. Send Request(Situation Handler);
if (Message. Type == Message Add Me Broad Cast)
Thread.SendRequest(AddNewDrone);if(Message.Type= =MessageAlive)Thread.SendRequest(UpdateDroneTable);
if (Message. Type == Message Re alignment)
Thread. Send Request(Realignment Drone);

4. Results and Discussion

4.1. Experimental Setup

To create drones' collaboration environment, AirSim, an open-source simulator developed by Microsoft, was used as a plug in for Unreal Engine [23]. We added a Flight Controller (FC) module to control the drone's movement programmatically and used MAVLINK message passing [20] pattern to communicate with the simulated drones. Computer Companion (CC) communicates with FC to receive sensor data and estimate drone states that can send commands for drone movement. The CC can control angles, velocity vector, destination position, or some combination of these by using move API [20]. We used image API to retrieve synchronized images from multiple cameras and applied a convolutional neural network for categorization. For experimentation, the proposed convolutional neural network was trained over 14000 images i.e., 4000 images for food requirement 4000 images for water requirement, 3000 images for clothing and 3000 images for the life-saving scenario. For coordination, AirSim API uses the North-East-Down (NED) coordinate system; the starting point of the vehicle is coordinated (0, 0, 0) in the NED system through settings. json which assigns geographic longitude, and altitude to the player's start component [23].

4.2. Simulation

To evaluate the proposed method, we generated different tasks in our simulating environment by putting images on the floor of the 3D environment. For simulation, the fleet of drones was prepared with Life-Saving, Food Delivery, Water Delivery, and Clothes Delivery drones with a count of 8, 7, 7, and 7, respectively. We mark the drones' as D1, D2, D3 and D4 with capabilities C1, C2, C3 and C4, respectively. When a task was generated, an alert appeared at that spot, and the drone categorized the images, taken through its camera.

We performed three experiments with an increasing number of tasks to measure the performance of a centralized system, the distributed system without enabling of realignment module, and the distributed system with realignment module. For each experiment, the number of messages passed to drones, the number of tasks completed, and the average time to complete a task were logged. In the first episode, we generated 100 tasks of different types, and the same experiment was repeated five times by an increase of 500 tasks each time. However, the capabilities remain the same in each experiment.

4.3. Experiment 1

In the first experiment, we implemented the scenario using a centralized system. Each drone was set to send situation information to the central server. The centralized system processed the results and assigned the task to the relevant drones. Once the drone completed its task, it sent feedback to the central server. The results of the experiment are given in Table 3.

Table 3. Results for centralized system.

Available Tasks	Task Completed	Total Number of Messages Passed	Completion Rate	Processing Time
100	82	164	82%	3 mins
600	470	940	78%	4 mins
1100	820	1640	75%	8 mins
1600	1100	2200	67%	16 mins
2100	1280	2560	60%	20 mins
2600	1430	2860	55%	32 mins

In this experiment, a situation identifying the drone sent the situation to the central server that processed it and instructed the same drone or any other drone to handle the situation. The centralized system has completed 82% of tasks during the first episode (Table 3 row 1). However, the completion rate gradually decreased with the increasing number of tasks. In the last episode, 2600 tasks were given, and the completion rate was reduced to 55%. The primary reason for this decline was the load on the server. The number of passed messages was almost twice as high as the number of completed tasks.

4.4. Experiment 2

In the second experiment, instead of sending the information to the central server, the drone directly communicated to the neighboring drones using the proposed solution. In this experiment, we have disabled the realignment module, which means that the two drones with the same capability can cover the overlapping area. The results are listed in Table 4.

Table 4. Results for distributed system without realignment.

Available Tasks	Task Completed	Total Number of Messages Passed	Completion Rate	Processing Time
100	70	280	70%	14 mins
600	495	1980	82%	12 mins
1100	940	3760	85%	08 mins
1600	1200	4800	75%	10 mins
2100	1370	5480	65%	15 mins
2600	1600	6400	62%	18 mins

In this experiment, the proposed distributed system presumed the same environment with the same number of tasks, but without its realignment module. This configuration allowed the drones to communicate with each other without the need for a central system. This scenario completed 70 % of the tasks in the first test (Table 4 row 1), and the performance increased as the number of tasks increased until the number of tasks being generated reached 1100. The performance started decreasing as the number of tasks increased, and it dropped to 62% for the last episode. However, the average task completion rate (73%) was better than that of the centralized system (69%).

4.5. Experiment 3

The third experiment was similar to experiment 2, except that the realignment module was enabled. Now, the two drones with the same capability do not overlap in the same coverage area. The results are logged in Table 5.

Table 5. Results for distributed system with realignment.

Available Tasks	Task Completed	Total Number of Messages Passed	Completion Rate	Processing Time
100	64	269	64%	13 mins
600	430	1806	72%	10 mins
1100	880	3696	80%	07 mins
1600	1350	5670	84%	04 mins
2100	1605	6741	76%	06 mins
2600	1800	7560	69%	12 mins

In this experiment, the same scenario was tested with a distributed setup having an active realignment module. The module allowed the drones to negotiate and occupy optimal positions within the space to increase the coverage area. At the start, the task completion rate was lower than either of the previous systems, but task completion rate started growing with the increasing number of jobs until the threshold point where the proposed system started giving better results.

A comparison of the proposed system's task completion rate with the existing solutions is given in Figure 5, whereas Figure 6 presents the message passing rate of the proposed system compared with the existing solutions.

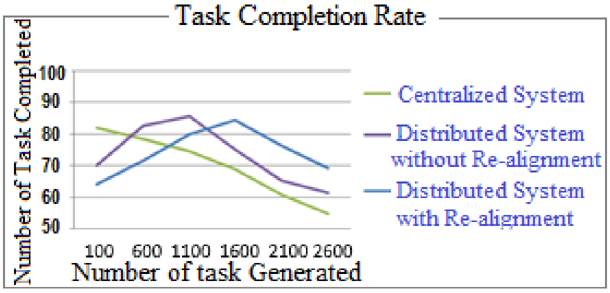

Figure 5. Comparison with respect to task completion rate.

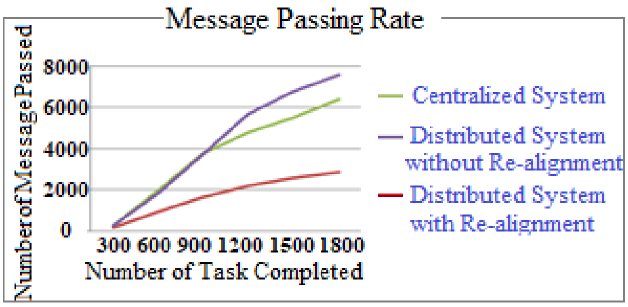

Figure 6. Comparison with respect to message passing.

As the number of drones was increased, the message passing increases that could result in a slow response time. Therefore, we suggest a set of fleet should not have more than 60 drones. For a wider coverage area, there could be multiple networks that can communicate with each other. In this way, we can keep local traffic within the same area, and when it is required, the message is sent to other networks.

Next, we present the performance analysis of our experimentation in the form of a confusion matrix. Five types of relief demanding situations were to be identified by the drones' fleet, including life-saving, food delivery, clothes delivery, and others (undefined). In Table 6, we logged the results of CNN-based classification as True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). Here, TP denotes the accurate prediction of a situation, FP denotes the wrongly predicted situations, FN denotes the other (undefined class) situations predicted wrongly into a defined class situation, and TN represents the other (undefined class) situations accurately predicted as other class situation.

Table 6. CNN Classification for Five types of relief demanding situations including life-saving, food delivery, clothes delivery, and others (undefined).

Actual Situation	Prediction				
	Life Saving	Food Delivery	Water Delivery	Clothes Delivery	Others
Life Saving	TP 2200	FP 300	FP 349	FP 100	FP 51
Food Delivery	FP 210	TP 3300	FP 290	FP 135	FP 65
Water Delivery	FP 275	FP 380	TP 3140	FP 165	FP 40
Clothes Delivery	FP 361	FP 200	FP 235	TP 2148	FP 56
Others	FN 57	FN 39	FN 37	FN 33	TN 46

These results of relief demanding situations' class prediction are then combined into a confusion matrix, given as Table 7. The accuracy, precision, recall, and F1-measure are then calculated for the outcomes and results are shown in Table 8.

Table 7. Confusion Matrix for prediction of target classes.

		Predicted Class	
		Class= defined	**Class= undefined**
Actual Class	Class= defined	TP 10788	FN 166
	Class= undefined	FP 3212	TN 46

Table 8. Confusion matrix based performance evaluations.

Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$	0.76
Precision	$\frac{TP}{TP+FP}$	0.77
Recall	$\frac{TP}{TP+FN}$	0.98
F1-Measure	$2*\frac{\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$	0.86

Accuracy being an intuitive performance measure, is simply a ratio of correctly predicted observation to the total observations. Our CNN-model got 76% accuracy, which is acceptable. The precision determines the percentage of correctly predicted positive observations, and a high precision relates to the low false-positive rate. We have got 0.77 pretty good precision. Recall measures the sensitivity of a model. It determines that what percent of the situation labeled as positive is actually positive. We got a recall score of 0.98, which is very good and shows that we achieved a high ratio of accurate positive predictions. F1 Score, being a weighted average of Precision and Recall, takes both false positives and false negatives into account. It is usually more useful than accuracy, especially if the cost of false positives and false negatives are very different. Our model achieved an F1 score of 0.86, which shows good performance has been achieved.

5. Conclusions

From the results, it is evident that the central server's performance drops as the number of tasks increases. The primary reason is the load on the server and its inability to resolve the tasks in parallel. A distributed system, without realignment, performed better than the centralized system. Still, it also suffers a low completion rate as the drones of similar capabilities tend to coexist in the same coverage area. On the other hand, the completion rate with the active realignment module is higher as it increased the coverage area. The work can be extended in the future using reinforcement learning and other AI techniques to improve reliability.

References

[1] Ahmad F., Kazmi S., and Pervez T., "Human Response to Hydro-Meteorological Disasters: A Case Study of the 2010 Flash Floods in Pakistan," *Journal of Geography and Regional Planning*, vol. 4, no. 9, pp. 518-524, 2011.

[2] Akhtar N. and Khan S., "Formal Architecture and Verification of A Smart Flood Monitoring System-of-Systems," *The International Arab Journal of Information Technology*, vol. 16, no. 2, pp. 211-216, 2019.

[3] Apvrille L., Roudier Y., and Tanzi T., "Autonomous Drones for Disasters Management: Safety and Security Verifications," *in Proceedings of 1st URSI Atlantic Radio Science Conference*, Las Palmas, pp. 1-2, 2015.

[4] Beaman B. and Ogata S., "Ultrastructural Analysis of Attachment to and Penetration of Capillaries in the Murine Pons, Midbrain, Thalamus, and Hypothalamus by Nocardia Asteroides," *Infection and Immunity*, vol. 61, no. 3, pp. 955-965, 1993.

[5] Bupe P., Haddad R., and Rios-Gutierrez F., "Relief and Emergency Communication Network based on an Autonomous Decentralized UAV Clustering Network," *in Proceedings of Southeast Con*, Fort Lauderdale, pp. 1-8, 2015.

[6] Choksi M., Zaveri M., Kumar J., and Pandey S., "Cloud-Based Real Time Data Acquisition in IoT Environment for Post Disaster Management," *in Proceedings of 9th International Conference on Computing, Communication and Networking Technologies*, Bangalore, pp. 1-6, 2018.

[7] Chouraqui S. and Selma B., "Unmanned Vehicle Trajectory Tracking by Neural Networks," *The International Arab Journal of Information Technology*, vol. 13, no. 6B, pp. 1020-2023, 2016.

[8] Chowdhury S., Emelogu A., Marufuzzaman M., Nurre S., and Bian L., "Drones for Disaster Response and Relief Operations: A continuous Approximation Model," *International Journal of Production Economics*, vol. 188, pp. 167-184, 2017.

[9] Dinesh M., Kumar S., Shetty S., Akarsh K., and Gowda K., "Development of an Autonomous Drone for Surveillance Application," *International Research Journal of Engineering and Technology*, vol. 5, no. 8, pp. 331-333, 2018.

[10] Drak A. and Asteroth A., "Autonomous Track and Follow UAV for Aerodynamic Analysis of Vehicles," *The International Arab Journal of Information and Technology*, vol. 16, no. 3A, pp. 587-93, 2019.

[11] Erdelj M., Krol M., and Natalizio E., "Wireless Sensor Networks and Multi-UAV Systems for Natural Disaster Management," *Computer Networks*, vol. 124, pp. 72-86, 2017.

[12] Erdelj M., Natalizio E., Chowdhury K., and Akyildiz I., "Help from the Sky: Leveraging UAVs for Disaster Management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24-32, 2017.

[13] Ganesh Y., Raju R., and Hegde R., "Surveillance Drone for Landmine Detection," *in Proceedings of International Conference on Advanced Computing and Communications*, Chennai, pp. 33-38, 2015.

[14] Gharibi M., Boutaba R., and Waslander S., "Internet of Drones," *IEEE Access*, vol. 4, pp. 1148-1162, 2016.

[15] Grippa P., Behrens D., Wall F., and Bettstetter C., "Drone Delivery Systems: Job Assignment and Dimensioning," *Autonomous Robots*, vol. 43, no. 2, pp. 261-274, 2019.

[16] Jeyaseelan A., *Satellite Remote Sensing and GIS Applications in Agricultural Meteorology*, World Meteorological Organisation, 2004.

[17] Kawamoto Y., Nishiyama H., Kato N., Ono F., and Miura R., "Toward Future Unmanned Aerial Vehicle Networks: Architecture, Resource Allocation and Field Experiments," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 94-99, 2018.

[18] Kumar J., Zaveri M., Kumar S., and Choksi M., *in Data and Communication Networks*, Springer, 2019.

[19] Ochoa S. and Santos R., "Human-Centric Wireless Sensor Networks to Improve Information Availability During Urban Search

and Rescue Activities," *Information Fusion*, vol. 22, pp. 71-84, 2015.

[20] Perl T., Venditti B., and Kaufmann H., "Ps Move Api: A Cross-Platform 6dof Tracking Framework," *in Proceedings of the Workshop on off-the-Shelf Virtual Reality*, Orlando, pp. 1-8, 2013.

[21] Rabta B., Wankmuller C., and Reiner G., "A Drone Fleet Model for Last-Mile Distribution in Disaster Relief Operations," *International Journal of Disaster Risk Reduction*, vol. 28, pp. 107-112, 2018.

[22] Ranjan A., Panigrahi B., Sahu H., and Misra P., "SkyHelp : UAV Assisted Emergency Communication in Deep Open Pit Mines," *in Proceedings of the 1st International Workshop on Internet of People, Assistive Robots and Things*, Munich Germany, pp. 31-36, 2018.

[23] Shah S., Dey D., Lovett C., and Kapoor A., "Airsim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," *in Proceedings Field and Service Robotics*, Zurich, pp. 621-635, 2018.

[24] Silva L., Bandeira R., and Campos V., "Proposal to Planning Facility Location Using UAV and Geographic Information Systems in a Post-Disaster Scenario," *International Journal of Disaster Risk Reduction*, vol. 36, 2019.

Zobia Zafar has completed her MS in Computer Science from the University of Engineering & Technology Lahore, Pakistan. She is working as a software developer and manager at Diabetes Management Centre Services Hospital, Lahore, Pakistan. Her research interests include distributed systems, drone applications, and disaster management.

Muhammad Awais has completed Ph.D. and MS Computer Science from the University of Engineering Technology, Lahore, Pakistan. He is currently working as an assistant professor at the Computer Science Department of the University of Engineering and Technology. His research interest includes Artificial Intelligence, Reinforcement Learning, Adaptive eLearning Systems, and Affective Computing.

Abdul Jaleel completed Ph.D. and MS in Computer Science from the University of Engineering Technology, Lahore, Pakistan. He is working as Assistant Professor and Head of the computer science department at Rachna College of the same University, in Gujranwala, Pakistan. His research interest includes developing self-managing software applications, autonomic computing, and software quality measurement metrics.

Fiaz Majeed completed Ph.D. degree in Computer Sciences from the University of Engineering and Technology, Lahore Pakistan, in 2016. Currently, he is serving as Head of Software Engineering Department under the faculty of Computing and Information Technology at the University of Gujrat (UOG), Pakistan. His research interests include Data Warehousing, Data Streams, Information Retrieval, and Social Networks.