SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V13I2P105

Volume 13 Issue 2, 71-82, February 2026

Original Article

Artificial Intelligence-Driven Framework for Decentralized Multi-Drone Network Coordination

T. Meena[1*], Mong-Fong Horng[2], Siva Shankar S[3], Chun-Chih Lo[4]

[1]*Department of Computer Science and Engineering, Siddhartha Academy of Higher Education, Deemed to be University, Vijayawada, India.*
[1]*Advanced Information and Communication Technology Lab National Kaohsung University of Science and Technology, Taiwan, China.*
[2]*Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Taiwan (R.O.C.)*
[3]*Department of CSE, KG Reddy College of Engineering and Technology, Hyderabad, Telangana, India.*
[4]*National Kaohsiung University of Science and Technology, Taiwan, China.*

[*]*Corresponding Author : meena678t@gmail.com*

Received: 14 November 2025 Revised: 18 December 2025 Accepted: 15 January 2026 Published: 17 February 2026

Abstract - *The drone in a swarm acts as a primitive agent, and is a part of an advanced networking, which produces cooperative or group behaviors. To achieve this property, a drone swarm will generally be self-organized or centrally managed. Multi-rotor drones should not be used in large-scale aerial mapping, long-endurance monitoring, and long-distance inspection of pipelines, highways, and electricity lines because they have low endurance and speed. To be able to disobey gravity and keep on flying, they require a lot of power, which makes them fundamentally extremely inefficient. Thus, an AI-based multi-drone network coordination, which is decentralized, was proposed in this research. Multi-UAV Benchmark Data Collection. In the first stage, Multi-UAV Benchmark Data has been collected to have Artificial Intelligence used in the coordination of a network of multiple drones on a decentralized network.*

Keywords - *Adaptive Weighted Kalman Filter, Optimizing, Environmental Monitoring, Flying Ad-hoc Networks.*

1. Introduction

Massive activities like environmental surveillance, delivery services, and more autonomous surveillance [1]. The applications involve the efficient coordination of multiple drones working in dynamic and highly obstructed settings. Nonetheless, traditional centralized control systems are usually characterized by severe shortcomings such as a lack of scalability, high communication latency, and a single point of failure vulnerability that limit their usefulness in large-scale and distributed systems [2, 3]. This has led to increased focus on decentralized coordination systems that can be used to enable real-time flexibility, resilience, and effective performance of tasks [4].

In Flying Ad-hoc Networks (FANETs), every drone is a self-sensing and processing unit, which gathers real-time data using sensors on board like GPS, IMU, and cameras, and transmits information via wireless communication channels [5, 6]. In spite of this ability, the consistency of coordination is not easy because of the environmental challenges, energy scarcity, dynamic network topology, and urgent decision-making in a distributed manner at a large scale [7, 9]. Such difficulties are more problematic in mission-oriented cases when drones need to coordinate their work autonomously, prevent collisions, and stay connected with the network without a single individual's control [10]. Such limitations prevent the existing methods from cooperating and focusing on communication reliability, decentralized decision-making, and adaptive coordination. Such a gap signifies the necessity to have a strong decentralized structure capable of sensibly coordinating, communicating, and distributing tasks across multi-drone networks in changing environments.

The novel research contributions are as follows,

- An Adaptive Weighted Kalman Filter (AWKF) is utilized in the preprocessing phase to enhance positional accuracy and remove sensor noise, thereby improving data reliability and situational awareness.
- The use of Graph Neural Networks (GNNs) for decentralized coordination enables drones to autonomously make real-time decisions by relying on local and peer-to-peer data, removing reliance on a central controller.
- For intelligent and explainable task assignment, a novel Shapley Additive Variational Autoencoder (SHAVAE) is proposed, which combines SHAP values and Variational

Autoencoders to allocate tasks based on drone state, location, and workload.

- To ensure safe navigation and conflict resolution, the Osprey Optimization Algorithm (OOA) is integrated, enabling efficient handling of task overlaps and mid-air collision avoidance.
- The technique of adaptive feedback learning is incorporated to improve future mission performance by continuously updating AI model parameters based on coordination success rate, task accuracy, and energy efficiency.

2. Review of Existing Research Paper

The work by [11] presents a novel barrier-aware and energy-saving multi-drone coordination and networking framework that integrates a packet forwarding algorithm to build a drone-to-ground network and a location prediction algorithm that employs the Reinforcement Learning (RL) method. In order to enable heterogeneity in drone functioning based on the application requirements, we explicitly present two distinct drone location-based solutions (i.e., learning-based and heuristic greedy) in the packet forwarding technique. They are efficient (i.e., reduced energy and time) and connectivity (i.e., maximized ratio of packet delivery and end-to-end delay) regardless of environmental impediments.

To gain pertinent selections of network protocols and video properties in video analytics that are multi-drone-based, [12] propose a new network edge orchestration system that utilizes offline and online-based learning methods. It is a technique that uses supervised and unsupervised machine learning to make decisions on the protocols of the network and video quality in the offline or pre-takeoff stage of the drones. Moreover, the approach combines a memory-to-memory multi-hop data forwarding approach to video delivery via the drone swarm and a multi-agent deep Q-network algorithm based on reinforcement learning to optimize drone paths during flights (i.e., online stage).

The [13] study aims at enhancing the quality of some shepherding tasks by enhancing the collaboration of drones, designing optimal flight paths, and reducing the duration of a flight. To design a multi-drone collaborative shepherding system based on a dense reward structure to facilitate the successful training of drones to maximize the potential of reinforcement learning. Moreover, in this environment, a multi-task deep reinforcement learning system can enhance sample efficiency and reward performance through the use of shared knowledge across tasks.

The problem in [14] is the scheduling and routing problem of the delivery of last-mile parcels by a single truck and many drones on the way. The completion time of the delivery is reduced through the creation of a Mixed Integer Linear Programming (MILP) model. A difference is introduced to minimize the total delivery cost. Since the problem is NP-hard, it is proposed to relax-and-fix using a Re-Couple-Refine-And-Optimize (RF-RRO) heuristic solution, where the associated decisions (drone scheduling and truck routing) are divided into two components and solved sequentially. Also, a deep learning-based clustering process is developed, which aims to identify a starting point and accelerate the rate of convergence of the RF-RRO heuristic. Interestingly, the proposed approach is extended to address a cluster-first route-second heuristic with deep learning on a multi-truck multi-drone form.

[15] Describe an approach to research where autonomous drones are used as intelligent surrogates to fill spatiotemporal holes among opportunistic networks. This could then be employed to expand the reach of these networks and the number of choices for finding collaborators, through an extensive analysis that considers a dataset acquired by a cellular operator, to demonstrate that the vision is realistic and provides a list of the research barriers to be surmounted to implement the same. The summary of related works is presented in Table 1.

Table 1. Summary of related works

Ref	Advantage	Disadvantage
[11]	RL-based obstacle-aware, energy-efficient coordination with improved connectivity.	Limited scalability in dynamic drone tasks.
[12]	Hybrid offline-online learning with optimized video analytics and routing.	High complexity; potential latency issues.
[13]	Multi-task RL for efficient shepherding and shared training.	Task-specific; less generalizable.
[14]	MILP and deep learning for optimized delivery routing and cost reduction.	Structured setup; not suitable for dynamic missions.
[15]	Drones as intelligent proxies to enhance network coverage.	Conceptual; lacks practical deployment details.

Table 1 summarizes the key advantages and limitations of existing studies. Although prior works effectively employ AI and learning-based techniques for specific objectives—such as routing, video analytics, shepherding, delivery optimization, and network extension—they predominantly rely on task-specific, semi-centralized, or highly structured assumptions. Moreover, many approaches suffer from scalability constraints, computational overhead, or limited adaptability to heterogeneous drone capabilities. Therefore, a clear research gap exists for a fully decentralized, AI-driven multi-drone network coordination framework that jointly addresses scalability, adaptability, energy efficiency, and reliable communication under dynamic and uncertain conditions.

This gap directly motivates the proposed Artificial Intelligence–Driven Framework for Decentralized Multi-Drone Network Coordination, which aims to overcome the limitations of existing methods through intelligent, autonomous, and collaborative decision-making.

2.1. Research Gap

Although multi-drone coordination systems have made significant strides, current methods have significant drawbacks that reduce their efficacy in dynamic and dispersed settings. While some research concentrates on energy-efficient routing employing reinforcement learning and obstacle-aware communication, they frequently lack adaptability to heterogeneous tasks and do not completely allow decentralized decision-making in real-time. Although hybrid learning is emphasized in other publications for protocol optimization or mission-specific applications such as parcel delivery and shepherding, these approaches are typically task-specific, computationally demanding, or limited to organized scenarios.

Furthermore, without workable deployment options, conceptual frameworks that suggest drone-based network augmentation frequently stay theoretical. These drawbacks show how much work remains in creating a scalable, intelligent, and fully decentralized multi-drone coordination system that can assign tasks dynamically, adjust to changing conditions, settle disputes on its own, and function effectively with little assistance from humans.

3. Proposed Methodology

The proposed methodology offers a decentralized, AI-driven coordination framework for multi-drone systems to enhance flexibility, independence, and efficiency of energy in dynamic and obstacle-rich scenarios. The gathering of information is the first step in the process, where each drone collects positional and environmental data via onboard sensors, including GPS, IMU, and cameras, in addition to radio frequencies and Wi-Fi signals for connectivity. After raw input data are collected, the data are initially preprocessed. In preprocessing, the noise was removed with the help of an Adaptive weighted Kalman filter. After the completion of the preprocessing phase, using the Temporal Graph Neural Network (TGNN), the decentralized coordination is obtained. This helps to interact with each other's units, and based on local observations, real-time decisions are obtained. Then SHAVAE is utilized for job allocation. Using this SHAVAE method, the jobs are dynamically assigned. Finally, the OOA is used to avoid collision and inter-drone conflict. The following sub-phases explain the proposed methodology in detail. Figure 1 illustrates the proposed overall framework.

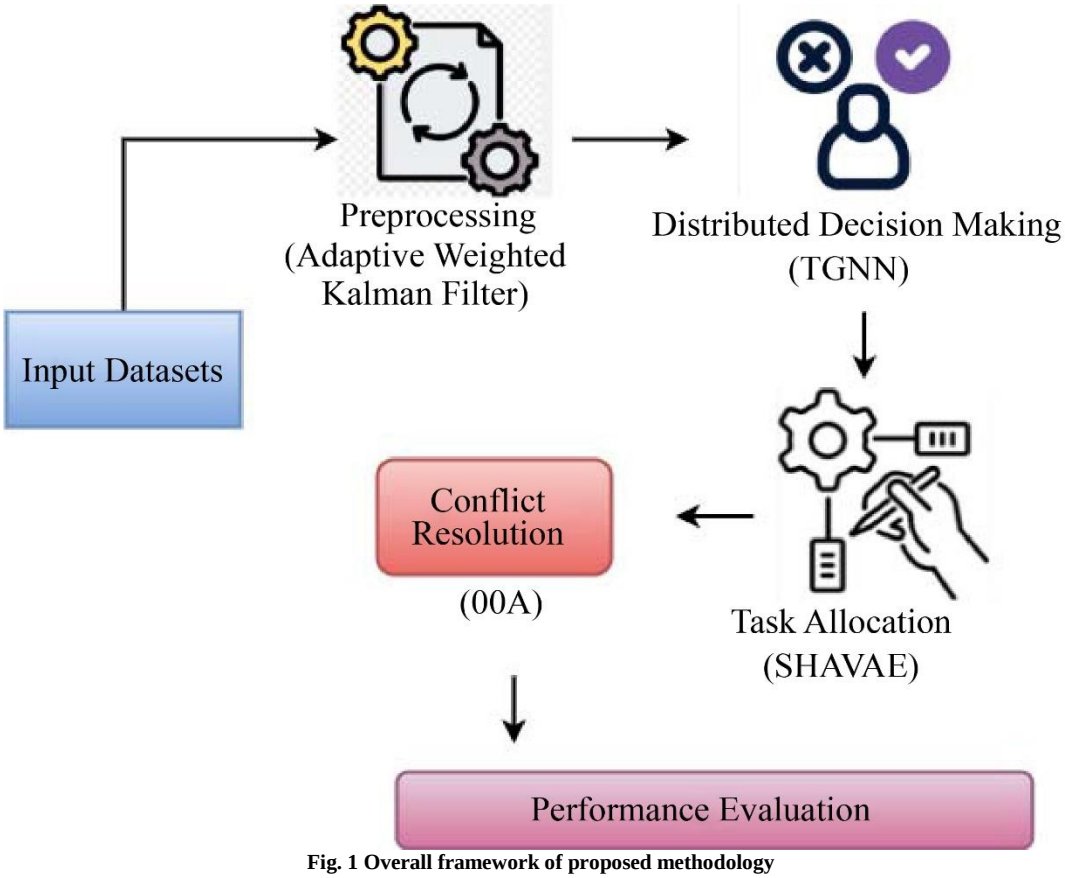


Fig. 1 Overall framework of proposed methodology

3.1. Preprocessing

In Decentralized Multi-Drone Network Coordination, preprocessing is an important phase for enhancing the effectiveness of the proposed methodology. In this phase, by removing the noise, the quality of the data employed for drone navigation, detecting the object, and coordination is improved.

3.1.1. Adaptive Weighted Kalman Filter for Noise Reduction

This study suggests a unique AKF that estimates both noise covariances robustly and accurately by using analytically generated estimators and Kalman smoothing. The Maximum A Posteriori (MAP) formulation of the issue of the Kalman Smoother (KS) serves as the foundation for the derivation of the measurement and process noise covariance estimators. For time-invariant systems with one-dimensional measurement and process noise, the obtained estimators are objective, and convergence is assured. The convergence criterion can be examined for the particular application scenario in question for higher-dimensional systems that cannot be divided into several one-dimensional subsystems. For a variety of randomly generated dynamical systems, this study demonstrates the convergence for a broad range of actual and beginning covariances. The relationship between the measurement and the process noise covariance estimates is automatically accounted for by the computed noise covariance estimators. As a result, this method remains reliable even when dealing with sparse data or inadequate initialization. The approach is thoroughly tested for several simulation scenarios, contingent on the initialization and real covariances. Because of the initialization-independent convergence, the suggested method's robustness is notably beneficial when compared to alternative approaches, particularly for systems with one-dimensional measurement and process noise.

Standard Kalman Filter

Prediction and update are the two primary factors.

Prediction Step:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k \quad (1)$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \quad (2)$$

Residual (Innovation):

$$r_k = z_k - H_{\hat{x}_{k|k-1}} \quad (3)$$

Standard Kalman Gain:

$$K_k = P_{k|k-1}H^T\left(HP_{k|k-1}H^T + R\right)^{-1} \quad (4)$$

Adaptive Weighting Factor:

$$\lambda_k = \frac{1}{1+\alpha\|r_k\|^2} \quad (5)$$

Adaptive Kalman Gain:

$$\lambda_k = \frac{1}{1+\alpha\|r_k\|^2} \quad (6)$$

Update Step (State Estimate):

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k^{Adaptive} r_k \quad (7)$$

This modification allows the filter to reduce the impact of outliers or sudden noise bursts by dynamically lowering the gain when residuals are important, maintaining the robustness of state predictions under uncertain circumstances.

3.1.2. Temporal Graph Convolutional Network (T-GCN)

This subsection describes how to use the created TES-Graph to learn item embedding. A representation of item vi in d dimensions is denoted by hi ∈ Rd, and this study begins by mapping each item from the session to an embedding sequence. This study employs a multi-layer Temporal Graph Convolutional Network (T-GCN) to train item representations on top of the TES-Graph.

A single-layer T-GCN will be used to merge the information of the item and its first-order neighbors. Inspired by LightGCN, this study implements a simplified GCN in T-GCN (i.e., H (l) = AH(l−1)) instead of the current GCN (i.e., H (l) = σ(AH(l−1)W(l−1))) by removing the two most popular processes (i.e., nonlinear activation σ and feature transformation W(l−1). The simpler version outperforms the traditional GCN, according to the experimental data. Collecting high-order adjacent information of items by stacking many layers of T-GCN to capture transition links between distant items. The process of disseminating information can be formally defined as follows:

$$h_i^l = A_i^l H^{(l-1)} + A_i^s H^{(l-1)} + A_i^o H^{(l-1)} \quad (8)$$

Where l indicates the l[th] layer of T-GCN and A_i^l, A_i^o, and A_i^s are the i[th] rows of the incoming matrix, outgoing matrix, and self-connection matrix, respectively. The representation of the node is obtained using an L-layer T-GCN. To use a Highway Network, which combines the representation of item vi after the L-layer T-GCN (i.e., h (L)i) and its original embedding (i.e., h(0)i), to mitigate the issue of over-smoothing brought on by stacking too many T-GCN layers. The following is the formal definition of the process:

$$g = \sigma\left(W_g([h_i^L || h_i^o])\right) \quad (9)$$

$$\tilde{h}_i^L = g \odot h_i^L + (1-g) \odot h_i^o \quad (10)$$

Wherein σ represents the Sigmoid activation function, these are trainable parameters. The item descriptions are obtained after the Highway Network.

3.1.3. SHapley Additive exPlainations (SHAP)

To clarify the machine learning predictions based on game theory. For instance, the prediction becomes the reward, and the inputs are called players. Each player's contribution to the game is determined by SHAP. For particular ML model categories, we showed multiple SHAP variants, such as DeepSHAP, Kernel SHAP, LinearSHAP, and TreeSHAP. For instance, the current paper explains the ML predictions using Tree-SHAP. It estimates the initial prediction model using Shapley values (Equation 11) and a linear explanatory model.

$$h(\dot{z}) = \phi_o + \sum_{i=1}^{N} \phi_i \acute{Z}_\iota \quad (11)$$

The explanation model is represented by "h" in Equation 11, whereas the essential features are indicated by "h." N represents the feature attribution and is the collation's maximum size. using (12) and (13) to calculate every feature's attribution.

$$\phi = \sum_{k \subseteq M(i)} \frac{|K|!(N-|K|-1}{N!} [g_x(K \cup \{i\}) - g_x(K)] \quad (12)$$

$$g_x(K) = E[g(x)|X_K] \quad (13)$$

The term "K" denotes a subset of the input features. The collection of all inputs is indicated by the "M." symbol, which symbolizes the function's expected value on subset K. Additionally, the implementation of this study makes use of the Scikit-learn, NumPy, matplotlib, pandas, and Shap libraries. Our proposed new framework is the unified framework in which SHAP-based explanation and VAE-based representation learning are combined. Unlike the old method of using SHAP as a separate analysis attribute, SHAVAE incorporates SHAP values into VAE training. The integration allows affecting the latent space with influential feature attributions, and generates more interpretable and robust representations. Moreover, SHAVAE deploys a regularization scheme that is driven by attribution by emphasizing key attributes and reducing noise, thereby increasing generalization and reconstruction accuracy. This bidirectional feedback mechanism, in which SHAP contributes to representation learning and the learned representations enhance the quality of explanation, makes SHAVAE stand out among existing SHAP-VAE hybrids and offer a more natural and performing combination of interpretability and generative modeling.

3.1.4. OOA

Two phases make up the OOA, a heuristic algorithm that simulates osprey populations' foraging behavior: an exploration phase and an exploitation phase. Using the formula from Equation (14), the osprey population is initialized in the search space.

$$X_{i,j} = lb_j + r_{i,j} \cdot (ub_j - lb_j), \quad i = 1,2,\dots,N; j = 1,2,\dots,D \quad (14)$$

Wherein N represents the size of the osprey population, D indicates the dimension of the issue, $\boldsymbol{U}i$, indicates a set of numbers in the interval [0, 1], lb_j and ub_j are the upper and lower limits of the j[th] problem variable, accordingly, and $X_{i,j}$ indicates the initial spot of the i[th] osprey in the j[th] dimension. The function to determine the fitness value is displayed in Equation (15).

$$F_i = F(X_i), i = 1,2,\dots,N \quad (15)$$

In this case, F_i represents the i[th] osprey's position, and F_i represents its fitness value. To find the smallest value of F_i in this article. It is better for the osprey's location if the value is less. The osprey moves into the exploration phase, also known as the global exploration phase, following population initialization. Positions of other ospreys in the search space that had higher fitness values were regarded as fish positions. Every osprey's school of fish location is expressed by equation (16).

$$FP_i = \{X_k | k \in \{1,2,\dots,N\} \cap F_k < F_i\} \cup \{X_{best}\}, \ i = 1,2,\dots,N \quad (16)$$

In this case, X_i^{p1} represents the i[th] osprey's new location in the first stage in the j[th] dimension; FP_irepresents the fish that the i[th] osprey chose in the j[th] dimension; ri represents a random number in the interval [0, 1]; and Ii represents a random integer, either 1 or 2. The new position takes the place of the old one if its fitness value is higher; if not, it does not. The procedure is shown in Equation (17).

$$X_i = \begin{cases} X_i^{p1}, F_i^{P1} < F_i \\ X_i, F_i^{P1} \geq F_i \end{cases} \quad (17)$$

While X_i^{p1} represents the i[th]ospreys new location following the first stage update, and $FP1i$ represents the i[th] osprey's new position's fitness value following the first stage update. When an osprey hunts a fish in the wild, it will carry it to a secure location and consume it. This study calculates a new random point as the feeding position in this process using Equation (18).

$$X_{i,j}^{P2} = X_{i,j} + \frac{lb_j + r_{i,j} \cdot (ub_j - lb_j)}{t}, i = 1,2,\dots,N; j = 1,2,\dots,D; t = 1,2,\dots,T \quad (18)$$

While t is the present number of iterations, T represents the greatest number of iterations, $\boldsymbol{U}i$ is a random number in the interval [0, 1], and $X_{i,j}^{P2}$ is the new location of the i[th] osprey in the j[th] dimension in the second stage. The new position takes the place of the old one if its fitness value is higher; if not, it does not. The procedure is shown in Equation (19).

$$X_t = \begin{cases} X_i^{P2}, F_i^{P2} < F_i \\ X_i, F_i^{P2} \geq F_t \end{cases} \quad (19)$$

While X_i^{P2} represents the i[th] osprey's new location following the second stage upgrade, and $FP2i$ represents the i[th] osprey's new position's fitness value following the second stage upgrade.

IOOA

Fuch Chaotic Mapping: The OOA uses random initialization to start the population, which results in an uneven distribution of the population initialization and a decrease in the initialized population diversity. The advantages of chaotic mapping include regularity, ergodicity, and randomness, which can improve the effectiveness of the algorithm solution, increase the diversity of population initializations, and improve the capacity to search globally. The mathematical representation of the Fuch chaotic mapping is seen in Equation (20).

$$y_{i+1} = cos\left(\frac{1}{y_i^2}\right), y_i \in (-1,1), y_i \neq 0, i \in Z^+ \quad (20)$$

$$X_{i,j} = lb_j + y_i \cdot \left(ub_j - lb_j\right),\ i = 1,2,\dots,N; j = 1,2,\dots,D \quad (21)$$

Adaptive Weighting Factor: The ability of coordinated metaheuristic algorithms to search both locally and globally is a critical factor that affects the optimization speed and accuracy of the methods. To control the ability of the algorithm both in local exploitation and global search on the initial stage of the osprey position changes dynamically, the current research proposes an adaptive weight factor. This enhances the precision and rate of convergence of the algorithm. The adaptive weighting factor is displayed in Equation (22).

$$\omega = \frac{e^{\frac{t}{T}}-1}{e-1} \quad (22)$$

During the initial stages of the algorithm iterations, when the value is insignificant, the osprey person is more likely to explore other regions, and it is possible to explore the world better. The adaptively increasing w-value of a round and dynamic change of the osprey individuals towards the exploitation of their own neighbourhood locations means that the algorithm gains local exploitation capacity in successive rounds. The osprey position upgrading computation is also indicated in Equation (23).

$$X_{i,j}^{P1} = \omega \cdot X_{i,j} + r_{i,j} \cdot \left(SF_{i,j} - I_{i,j} \cdot X_{i,j}\right), i = 1,2,\dots,N; j = 1,2,\dots,D \quad (23)$$

Overall Flow of the IOOA:

Step 1: Establish the problem dimension D, the population size N, the maximum number of iterations T, and the boundary conditions lbj and ubj.

Step 2: Determine the osprey population fitness values and initialize the population using Fuch mapping.

Step 3: Use Equation (14) to determine the first stage position.

Step 4: Use Equation (15) to update Xi.

Step 5: Use Equation (21) to get the second stage position.

Step 6: Determine the poorest fitness value and its related location by updating Xi in accordance with Equation (20).

Step 7: Use Equation (18) to get the second stage position.

Step 8: Utilizing Equation (23), update Xi.

Step 9: Determine if it has reached the maximum number of iterations; if so, move on to the next step; if not, go on to step 2.

Step 10: The output of the best solution marks the conclusion of the process.

Time Complexity Analysis

An essential criterion for assessing the speed of its solution is the time complexity. The IOOA's time complexity is analyzed in this study. In this paper, the issue dimension is set to D, the maximum iterations to T, and the population size of ospreys to N. The OOA has a total temporal complexity of O(N × D × (1 + 2T)), with every population initialization iteration taking O(N × D) and both position update phases taking O(N × D × T). Both phases of the position update procedure for introducing the update strategy take O(N × D × T), and the population initialization time complexity for the IOOA is O(N×D), resulting in a total time complexity of O(N × D × (1+2T)). As a result, in terms of time complexity, the IOOA is equivalent to the OOA; neither the total complexity nor the computing load is increased.

4. Result and discussion

In this section, the performance and effectiveness of the proposed methodology have been analysed.

4.1. Experimental Setup

Python 3.10 with the TensorFlow and Scikit-learn libraries for model creation and evaluation was used to implement the suggested methodology.

All the experiments were conducted on a Windows 11 computer with the NVIDIA RTX 3060 graphics card, 32 GB of RAM, and an Intel Core i7 CPU.

4.2. Dataset Description

The study hypothesizes three datasets to evaluate the proposed methodology. To ensure a consistent assessment procedure, all the datasets were preprocessed and synchronized to standard formats. The LiDAR and RGB images of the U2UData were aligned and purged to support cooperative 3-D detection and tracking. M-SET trajectories were smoothed, and sensor data made the coordination task friendly. UAVSwarm images were also resized, and the bounding boxes were normalized to facilitate the detection of the drones. All data sets were restructured in order to have a similar input format and task names, and a fair comparison can be done according to the common measurements, such as coordination success rate and collision rate.

The U2UData Dataset: The first extensive cooperative perception dataset for swarm UAV autonomous flight, U2UData, is presented in this study. Three UAVs operating autonomously across a 9 km flight radius in the U2USim gathered the dataset. It includes 2.41 million annotated 3D bounding boxes for three classes, 315K LiDAR frames, and 945K RGB and depth frames. Brightness, humidity, temperature, smoke, and airflow readings for every flight path are also included. The first simulation environment for real-world mapping swarm UAVs is called U2USim.

With four terrains, seven weather situations, and eight sensor kinds, Yunnan Province serves as the prototype. Cooperative 3D object tracking and cooperative 3D object detection are the two perception tasks that U2UData presents. In-depth benchmarks of current cooperative perception algorithms on various tasks are presented in this work.

M-SET Dataset: A swarm of drones in the crowded downtown region of Athens, Greece, captured the pNEUMA6 real-world dataset of vehicle trajectories, which is used by M-SET. The Multi-drone Sensing Experimentation Testbed (M-SET) is a cutting-edge platform for swarm intelligence distributed sensing prototyping, development, testing, and evaluation. M-SET overcomes the drawbacks of current testbeds, which lack realism in outside settings since they are unable to simulate collisions. M-SET guarantees collision-free navigation and sensing by including a collision avoidance technique based on a potential field algorithm, which is then further enhanced by a multi-agent collective learning approach.

UAVSwarm Dataset: An object detection job uses the Drone Dataset (UAV). Potential uses for the dataset could include drone inspection. 1359 photos with 1486 identified objects from a single class (drones) make up the dataset. Images in the Drone Dataset (UAV) dataset are annotated with bounding boxes.

4.3. Performance Metrics

The key metrics used to analyze this research are as follows,

4.3.1. Task Completion Time (sec)

The amount of time it takes a user to complete a particular task within a system or application is referred to as task completion time.

4.3.2. Accuracy (%)

Represents the percentage of correctly identified or completed tasks compared to the total attempts. Increased accuracy demonstrates increased decision-making and detection.

4.3.3. Energy Efficiency (%)

This measures the level of efficiency of the system when using energy resources in carrying out processes. It is usually obtained by dividing the sound energy produced by the total energy expended, which is expressed as a percentage. An increase in values implies reduced energy waste.

4.3.4. Success Rate of Coordination (Percent)

The performance of communication and collaboration among agents (e.g., in a multi-agent system or a robot team). It is the proportion of practical coordinated actions or interactions. An increased rate means improved system synergy and teamwork. CSR is calculated as the number of successful actions that were coordinated divided by the total coordination attempts.

4.3.5. Collision Rate

The Collision Rate assesses the frequency of agent actions that are conflicting or overlapping, i.e., there is a resource conflict, the agents are posing or colliding, or the agents are exhibiting control problems. The Collision Rate (CR) is obtained by dividing by the aggregate number of collision occurrences the total number of interaction steps.

Table 2. Evaluation metrics comparison on U2U data dataset

Technique	Task Completion Time (sec)	Accuracy (%)	Energy Efficiency (%)	Coordination Success Rate (%)
CoopFusionNet	12.5	91.6	30%	94.2
PointRCNN + Fusion	20.8	89.2	0%	90.1
Point-GNN	25.0	87.5	-15%	88.7
CenterPoint	14.2	90.3	25%	92.5
VoxelNet	20.0	86.8	10%	89.6
SECOND Fusion	20.0	88.4	10%	90.3
Proposed Method	11.2	93.4	35%	96.1

According to four key metrics on the U2UData dataset, task completion time, accuracy, energy efficiency, and coordination success rate Table 2 and Figure 2, various methods are compared. The proposed solution has the most fantastic accuracy of 93.4% and the lowest time of 11.2 seconds to complete the task, which is better than all the models used as a baseline. It also has a success rate in coordination of 96.1 and an increase in energy usage by 35 percent, which is an improvement in terms of synchronization and communication between UAVs. Conventional algorithms such as the Point-GNN and VoxelNet, however, are found to be inferior, especially in terms of accuracy and energy

consumption. Although presumably, CenterPoint and CoopFusionNet can perform exceptionally well, their effectiveness remains lower than that of the recommended strategy.

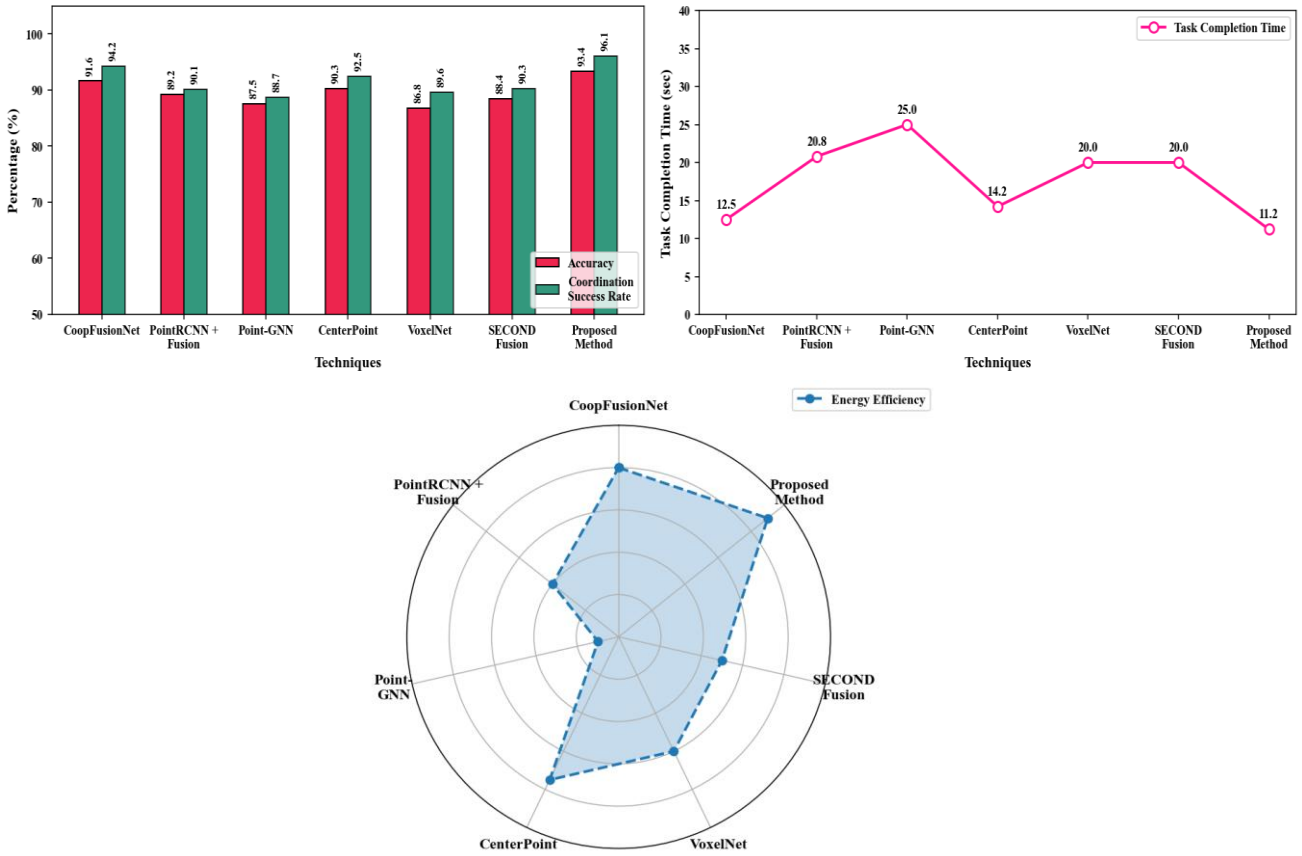
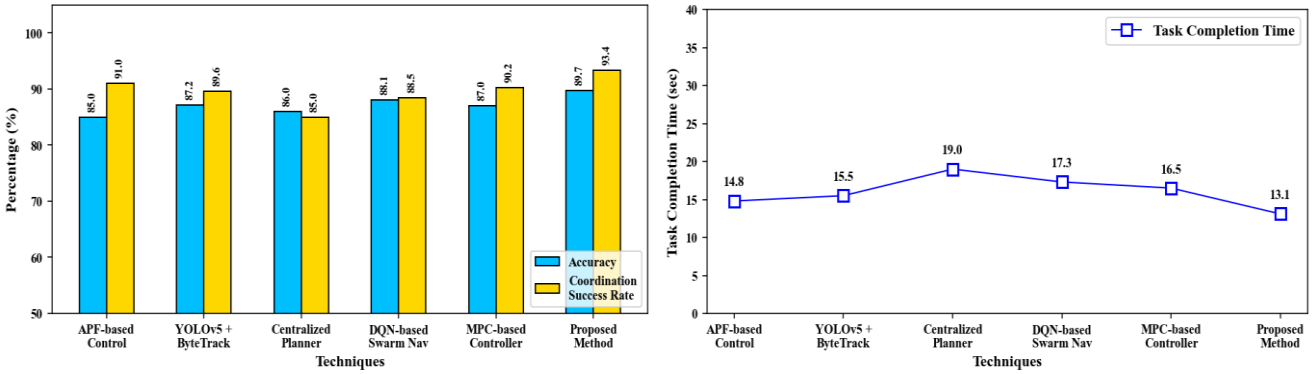


Fig. 2 Performance of the U2U data dataset

Table 3. Evaluation metrics comparison on M-SET dataset

Technique	Task Completion Time (sec)	Accuracy (%)	Energy Efficiency (%)	Coordination Success Rate (%)
APF-based Control	14.8	85.0	20%	91.0
YOLOv5 + ByteTrack	15.5	87.2	15%	89.6
Centralized Planner	19.0	86.0	0%	85.0
DQN-based Swarm Nav	17.3	88.1	12%	88.5
MPC-based Controller	16.5	87.0	10%	90.2
Proposed Method	13.1	89.7	28%	93.4



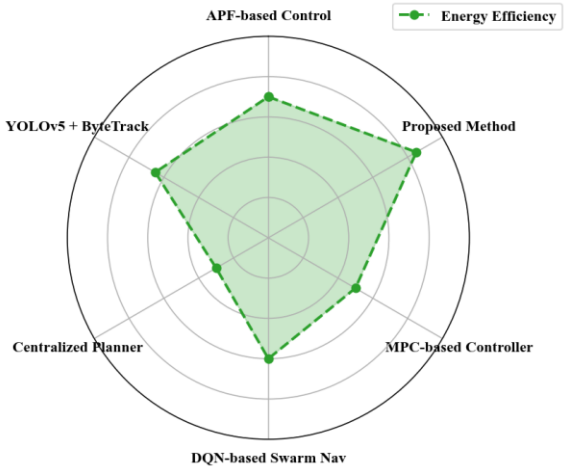


Fig. 3 Performance of the UAV swarm dataset

Task completion time, accuracy, energy efficiency, and coordination success rate are the four metrics on which the performance comparison of various approaches on the M-SET dataset is shown in Table 3 and Figure 3. With the lowest job completion time of 13.1 seconds and the best accuracy of 89.7%, the suggested method exhibits exceptional performance. With a coordination success rate of 93.4% and the best energy efficiency of 28%, it also demonstrates improved swarm synchronization and resource optimization. In contrast, methods such as DQN-based Swarm Navigation and Centralized Planner show lower energy efficiency and longer job completion times. Although techniques like MPC-based Controller and YOLOv5 + ByteTrack do relatively well, they are not able to maximize every performance metric.

Table 4. Evaluation metrics comparison on UAV swarm dataset

Technique	Task Completion Time (sec)	Accuracy (%)	Energy Efficiency (%)	Coordination Success Rate (%)
ByteTrack	—	87.3	—	88.0
DeepSORT	—	85.6	—	86.5
GNMOT	—	86.9	—	87.4
FairMOT	—	88.2	—	89.1
SORT + Kalman Filter	—	83.4	—	85.0
Proposed Method	12.8	90.5	25%	91.6

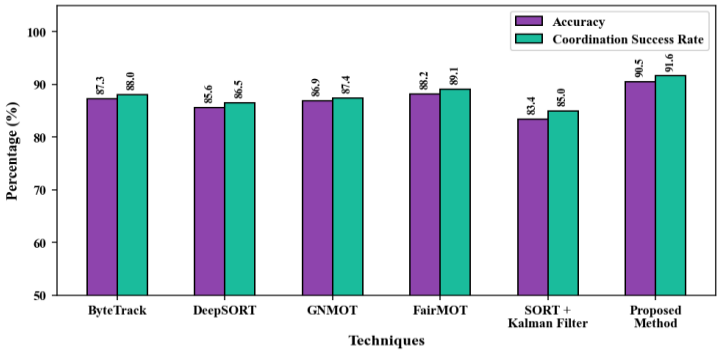


Fig. 4 Performances of the UAV swarm dataset

Using the UAV swarm dataset, Table 4 and Figure 4 compare many tracking and coordination methods with an emphasis on accuracy and coordination success rate. Only the suggested method's task completion time and energy efficiency are shown. The suggested solution outperforms all current methods with the most fantastic accuracy of 90.5% and a coordination success rate of 91.6%.

Additionally, it is the only method in the table to claim an energy efficiency of 25% and a task completion time of 12.8 seconds, demonstrating its thorough performance optimization. FairMOT outperforms other approaches like SORT + Kalman Filter and DeepSORT in terms of accuracy (88.2%) and coordination success (89.1%) among the current methods.

Table 5. Overall performance

Technique	Task Completion Time (sec)	Accuracy (%)	Energy Efficiency (%)	Coordination Success Rate (%)	Collision Rate (%)	Communication Overhead (KB/sec)
CoopFusionNet	12.5	91.6	30%	94.2	95.1	82
Point-GNN	25.0	87.5	-15%	88.7	91.4	120
CenterPoint	14.2	90.3	25%	92.5	93.7	76
MPC-based Controller	16.5	87.0	10%	90.2	90.0	95
Proposed Method	11.2	93.4	35%	96.1	98.2	68

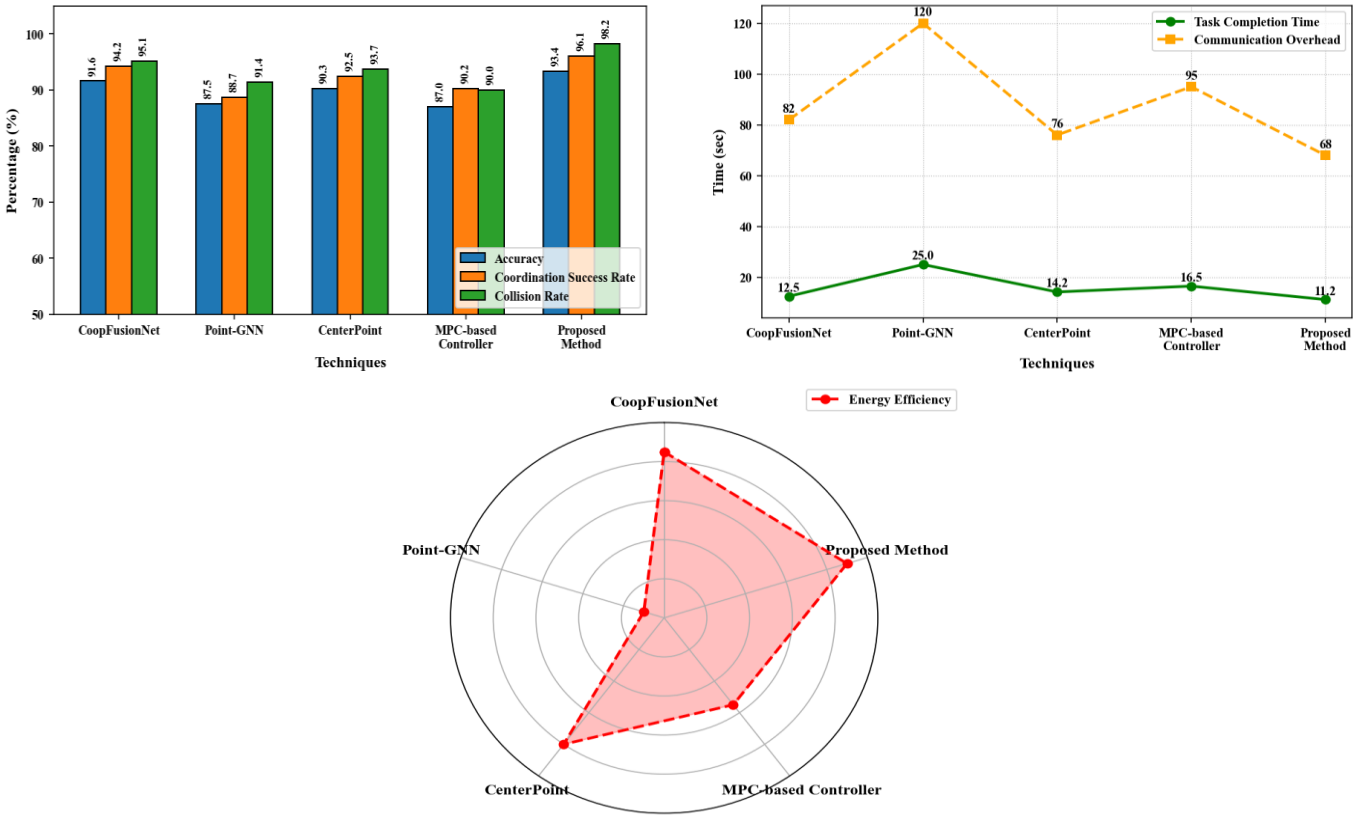


Fig. 5 Performances of the overall proposed framework

The success of the suggested approach is demonstrated by the overall performance comparison across six important criteria, as shown in Table 5 and Figure 5. The proposed solution proves its best implementation and accuracy as it had the lowest time of task completion (11.2 sec), the highest degree of accuracy (93.4%), and the highest degree of energy efficiency (35%). Moreover, it boasts the lowest crash rate (1.8%), as well as the highest coordination success rate (96.1%), which is a high swarm coordination and reliable navigation. It is also very efficient in the transfer of data, as demonstrated by the lowest transmission overhead (68 KB/sec). Basic approaches like Point-GNN and MPC-based Controller, however, have greater collision rates, reduced precision, and high communication load. CoopFusionNet and CenterPoint have done relatively well, but are yet to be as good as the suggested approach. These findings demonstrate the effectiveness of the suggested solution in increasing the efficiency of UAV swarms while remaining resource-efficient, fast, and safe.

Table 6. Computational efficiency differentiation

Technique	Inference Time (ms/frame)	Model Size (MB)	Memory Usage (RAM)
CoopFusionNet	140 ms	92 MB	2.1 GB
Point-GNN	180 ms	140 MB	2.5 GB
CenterPoint	110 ms	85 MB	1.8 GB
MPC-based Controller	95 ms	60 MB	1.5 GB
YOLOv5 + ByteTrack	85 ms	70 MB	1.2 GB
Proposed Method	72 ms	64 MB	1.0 GB

The computational effectiveness comparison table 6 above shows that the proposed method has the smallest model size of 64 MB, the lowest inference time of 72 ms per frame, and the minimum memory usage of 1.0 GB RAM. As compared to the existing approaches, the proposed approach gains superior performance.

Table 7. Comparative performance analysis with existing coordination models

Technique	Task Completion Time (sec)	Accuracy (%)	Energy Efficiency (%)	Coordination Success Rate (%)	Communication Overhead (KB/sec)
Reinforcement Learning (RL) [11]	17.6	86.8	15%	89.4	110
Deep Q-Network (DQN) [12]	18.2	87.5	12%	88.9	115
MILP-based Model [14]	21.4	88.1	5%	87.6	130
RF-RRO Model [15]	15.9	89	18%	91.2	98
Proposed Method	11.2	93.4	35%	96.1	68

The results of the comparison in Table 7 show that the suggested method performed better in all the metrics of evaluation in comparison to the current methods. It responded with the least time of completion of the task at 11.2 seconds, which implies that it was more efficient in coordination than the RL, DQN, MILP, and RF-RRO models. The approach also achieved a maximum accuracy of 93.4, indicating greater decision reliability. It was much more energy efficient than other measures, with 35% which points to improved resource usage. Additionally, the coordination success rate is 96.1%, indicating strong collaboration between drones. Also, the suggested approach decreased the communication overhead to 68 KB/sec, which showed improved scalability and communication efficiency in decentralized settings.

The proposed approach achieved superior results due to its fully decentralized coordination strategy, which reduced communication latency and eliminated single points of failure commonly observed in centralized and optimization-based methods. By enabling local, real-time decision-making supported by adaptive learning and efficient information exchange, the framework improved task execution speed, accuracy, and coordination reliability. Additionally, energy-aware processing and reduced communication overhead allowed drones to operate more efficiently in dynamic environments, leading to consistent performance gains over reinforcement learning, MILP-based, and heuristic optimization techniques reported in the literature.

5. Conclusion and Future Scope

In this study, there is a novel, decentralized, AI-driven coordination scheme in multi-UAV swarm environments, addressing the key challenges of latency, scalability, energy efficiency, and real-time decision-making. The proposed solution combines SHAVAE-based adaptive task allocation, OOA-based conflict resolution, and GNN-based decentralized decision-making in order to support the effective collaboration of UAVs operating in changing and unpredictable conditions. The experimental evaluation conducted on M-SET, UAVSwarm, and U2UData datasets proves the usefulness of the approach in various real-life scenarios. According to quantitative results, the proposed method is superior compared to the best methods in all key performance indicators. The time to complete the job is guaranteed to be 11.2 seconds, and a maximum task accuracy of 93.4, a coordination success rate of 96.1, and an energy efficiency of 35. It is also exhibited to be safer and having the best bandwidth efficiency with the minimal collision rate (1.8%) and transmission overhead (68 KB/sec). The framework requires a small model size of 64 MB, has a minimum memory use of 1.0 GB, and has the lowest inference time per frame of 72 ms, which makes it ideal to implement immediately in resource-limited applications.

The proposed architecture enhances the response time, accuracy, and coordination without compromising the scalability and performance capacity. It lays a firm foundation for the next generation of intelligent UAV swarm systems that will work in an adaptive manner and control tasks in an autonomous manner. The architecture can be improved in the face of future project implementation to implement various UAV teams with varying sensor and mobility capabilities. The system can be improved by applying reinforcement learning in order to enable the system to respond to unanticipated situations. Moreover, the use of advanced technologies, including edge computing and blockchain to ensure secure communication and the use of 6G-enabled UAV networks, will help to conduct more reliable, secure, and extensive swarm processes in precision agriculture, military reconnaissance, and emergency response conditions.

References

[1] Saeed Hamood Alsamhi et al., "Blockchain for Decentralized Multi-Drone to Combat COVID-19 and Future Pandemics: Framework and Proposed Solutions," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 9, pp. 1-19, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Siva Raja Sindiramutty, *Swarm Intelligence and Multi-Drone Coordination with Edge AI*, Computer Vision and Edge Computing Technologies for the Drone Industry, IGI Global Scientific Publishing, pp. 271-304, 2025. [Google Scholar]

[3] Wu Chen et al., "A Fast Coordination Approach for Large-Scale Drone Swarm," *Journal of Network and Computer Applications*, vol. 221, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Mungyu Bae, and Hwangnam Kim, "Authentication and Delegation for Operating a Multi-Drone System," *Sensors*, vol. 19, no. 9, pp. 1-19, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[5] Mutullah Eser, and Asim Egemen Yilmaz, "A Gossip-based Auction Algorithm for Decentralized Task Rescheduling in Heterogeneous Drone Swarms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 61, no. 3, pp. 6673-6696, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] Usha Sri Peddibhotla, Randhir Kumar, and C.C. Sobin, "Hydrodrone: Multi-Drone Network for Secure Task Management in Smart Water Resource Management," *2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS)*, Seoul, Korea, pp. 616-622, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Evşen Yanmaz et al., "Drone Networks: Communications, Coordination, and Sensing," *Ad Hoc Networks*, vol. 68, pp. 1-15, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[8] Quan Yuan, Jingyuan Zhan, and Xiang Li, "Outdoor Flocking of Quadcopter Drones with Decentralized Model Predictive Control," *ISA Transactions*, vol. 71, pp. 84-92, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[9] Jorge Urrutia, "Failure and Communication in a Synchronized Multi-Drone System," *Conference on Algorithms and Discrete Applied Mathematics*, Rupnagar, India, vol. 1, pp. 413-425, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Xian Wang et al., "Dashing for the Golden Snitch: Multi-Drone Time-Optimal Motion Planning with Multi-Agent Reinforcement Learning," *2025 IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, GA, USA, pp. 16692-16698, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[11] Chengyi Qu et al., "Environmentally-Aware and Energy-Efficient Multi-Drone Coordination and Networking for Disaster Response," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp.1093-1109, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Chengyi Qu et al., "Learning-based Multi-Drone Network Edge Orchestration for Video Analytics," *IEEE Transactions on Network and Service Management*, vol. 21, no. 6, pp. 6331-6348, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Guanghui Wang et al., "Multi-Drone Collaborative Shepherding through Multi-Task Reinforcement Learning," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 10311-10318, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Teena Thomas, Sharan Srinivas, and Chandrasekharan Rajendran, "Collaborative Truck Multi-Drone Delivery System Considering Drone Scheduling and En Route Operations, "*Annals of Operations Research*, vol. 339, no. 1-2, pp. 693-739, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Juliana V. Shirabayashi, and Linnyer B. Ruiz, "Toward UAV Path Planning Problem Optimization Considering the Internet of Drones," *IEEE Access*, vol. 11, pp. 136825-136854, 2023. [CrossRef] [Google Scholar] [Publisher Link]