Transfer Learning-Enabled System for Drone Medicine Delivery Based on Spatio-Temporal Remote Sensing Data in Edge Cloud Networks

Abdullah Lakhan, Tor-Morten Grønli, *Member, IEEE*, Ahmet Soylu, Ghulam Muhammad, *Senior Member, IEEE*, Qurat-ul-ain Mastoi, *Member, IEEE*, and Huaming Wu, *Senior Member, IEEE*

***Abstract*—These days, satellite remote sensing data is employed for different drone applications. The main goal is to provide imaginary information about electromagnetic locations and patterns, along with insights into geolocations on Earth. The Internet of Drone Things (IoDT) exploits remote sensing data to deliver medicine from source to destination. However, many existing medicine delivery systems based on drones need longer execution times and more efficiency in delivering medicine to the right destinations. This paper presents transfer learning, which empowers a spatiotemporal remote sensing data training system for medicine delivery in edge cloud networks based on IoDT applications. The objective is to deliver the medicine to the original destination with the highest score and process all drone tasks based on their given deadlines. We present the offloading spatiotemporal training and scheduling (OSPTS) algorithm methodology that completes the data collection process and medicine delivery in different locations. Therefore, we solve the problem as a combinatorial problem and find the optimal solution based on searching and convolutional neural networks (CNN). Transfer learning and convolutional neural networks are sub-schemes of the OSPTS that train the remote sensing data on edge nodes and point clouds for optimal medicine delivery. Simulation results show that the OSPTS obtained the highest score for medicine delivery in the correct position with less processing time than existing systems.**

***Index Terms*—Remote sensing data, Internet of drone things (IoDT), spatio-temporal, transfer learning, convolutional neural networks.**

Received 25 November 2024; revised 8 September 2025; accepted 27 November 2025. Date of publication 1 December 2025; date of current version 10 March 2026. This work was supported in part by the Natural Science Foundation of Tianjin under Grant 25JCYBJC01540, in part by the Emerging Frontiers Cultivation Program of Tianjin University Interdisciplinary Center, and in part by the Ongoing Research Funding Program, King Saud University, Riyadh, Saudi Arabia, under Grant ORF-2025-34. Recommended for acceptance by F. Ye.*(Corresponding author: Huaming Wu.)*

Abdullah Lakhan is with the Department of Cybersecurity, Dawood University of Engineering and Technology, Karachi 74800, Pakistan (e-mail: abdullah.lakhan@duet.edu.pk).

Tor-Morten Grønli and Ahmet Soylu are with the School of Economics, Innovation, and Technology, Kristiania University of Applied Sciences, Prinsens, 70153 Oslo, Norway (e-mail: Tor-Morten.Gronli@kristiania.no; ahmet.soylu@kristiania.no).

Ghulam Muhammad is with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia (e-mail: ghulam@ksu.edu.sa).

Qurat-ul-ain Mastoi is with the School of Computer Science Creative Technologies, University of the West of England, BS16 1QY Bristol, U.K. (e-mail: qurat-ul-ain.mastoi@uwe.ac.uk).

Huaming Wu is with the Center for Applied Mathematics, Tianjin University, Tianjin 300072, China (e-mail: whming@tju.edu.cn).

Digital Object Identifier 10.1109/TCC.2025.3639073

I. Introduction

ARTIFICIAL intelligence (AI) has been gaining popularity to make systems more automatic and realistic with boosted performance. Meanwhile, AI exhibits optimal performance in spatiotemporal sequence prediction for various applications such as environmental monitoring, disaster management, medicine delivery, drone flying, urban planning, and aircraft for vegetation indices [1]. These applications generate remote sensing data by connecting services to satellites and placing them in geographical areas within networks. Many aircraft sensors are mounted on different drones and offload and download remote sensing data in collaborative learning from other satellites during their journey at various locations and times [2]. AI algorithms such as generative adversarial network can forecast patterns and sequences in remote sensing data from apps based on historical data and determine future events for various applications [3]. The Internet of Drone Things (IoDT) is an emerging technology that provides multiple services in different areas, including surveillance, traffic anomaly detection in aerial videos, atmospheric reviers, location searching, disaster relief, and food delivery [4], [5].

During the pandemic, remote sensing data-enabled drone medicine delivery systems were implemented in urban cities [6], [7], [8], [9]. Deep learning and machine learning with computer vision techniques are implemented to record and store the sensing data of bean cultivation, parcel delivery, traffic flaws for decision-making. IoDT drones have many remote sensing applications, e.g., crops and others, which are implemented at the edge and cloud nodes, similar to previous studies [10], [11], [12], [13], [14], [15]. Historical data such as forest, weather, regional, and flood risk are collected to predict paths, locations, and times with different objectives. The data has different diversity, e.g., location, time, video, images, and numeric data, which is collected from different drone sensors, satellites, edge nodes, and cloud nodes for remote sensing processing. Therefore, due to their multi-feature, non-linear data, deep learning has boosted the performance of applications to extract features with more accuracy.

TABLE I
EXISTING WORKS ON REMOTE SENSING APPLICATIONS

Paper	Data	Methods	Nodes	Objective	Score
[1]	Spatiotemporal	DL	Node	Find Path	✗
[2]	Fusion Spatio	ML	Node	Min.Delay	✗
[3]	Fusion Spatio	TL	Node	Accuracy	✗
[4]	Spatio-temporal	DL	Node	Security	✗
[5], [6]	Spatio Temporal	CNN	Node	Accuracy	✗
[6]	Fusion Spatio	TL	Cloud	Accuracy	✗
[7]	satellite remote	CNN	Node	Time	✗
[8]–[10]	Location Spatio	ML	Node	Accuracy	✗
[11], [16]	Data	ML	Node	Accuracy	✗
[12], [13]	Locations	ML	Node	Accuracy	✗
[14], [15]	Land Angles	ML	Node	Accuracy	✗
[17], [18]	Fusion Images	ML	Node	Accuracy	✗
[19]	UAV	RFL	Node	Accuracy	✗
[20]	Spatio Temporal	RFL	Node	Accuracy	✗
Proposed	Fusion	Many	Many	Many	✓

However, existing systems pose research challenges or questions. The paper addresses two main challenges. Firstly, the current AI-enabled drone medicine delivery system only prioritizes accuracy and neglects time, deadlines, and resource constraints related to sensing nodes. Secondly, placing sensing nodes, such as satellites and others, requires a more efficient application approach.

In this paper, we solve the challenges above and suggest enhancing spatiotemporal sequence prediction for medicine delivery on the IoDT using transfer learning of convolutional neural networks (CNN) with remote sensing data. We consider the multi-modal remote sensing data with the spatio-temporal features from IoDT and satellite nodes. We aim to improve the delivery pattern and minimize the time for IoDT applications in the proposed framework. Therefore, with the multi-constraint problem, we formulate this problem combinatorially with convex and concave optimization. The paper makes the following contributions:

- We present enhanced spatiotemporal sequence prediction for a medicine delivery framework on the IoDT using transfer learning convolutional neural networks with remote sensing data.
- For the combinatorial problem, we present the offloading spatio-temporal training and scheduling (OSPTS) algorithm methodology that completes the data collection process and medicine delivery in different locations.
- We present transfer learning for training multi-modal spatiotemporal remote sensing data in medicine delivery drone applications.
- We show a technique to use an explainable convolutional neural network to train remote sensing data for the best pattern and sequence prediction for the IoDT application requirements.

The paper is organized in the following way: Section II is about related work. Section III defines the problem formulation. Section IV defines the methodology. Section V is the performance evaluation and result analysis. Section VI is the conclusion and future work.

II. RELATED WORK

This section discusses spatio-temporal remote sensing data for sequence and pattern prediction in drone applications across various cases. The cases are crop and medicine delivery, where the data is trained on deep learning and AI algorithms.

We defined Table I to describe the existing studies with data, methods, applications, nodes, objectives, and limitations. We utilised various abbreviations, including machine learning (ML), deep learning (DL), reinforcement learning (RL), and transfer learning (TL), in conjunction with uncrewed aerial vehicles (UAVs) in their studies. The work in [1] presents a forecasting vegetation and used deep learning and AI algorithms for sequence and pattern detection based on spatiotemporal remote sensing data in nodes. However, this work only discusses the abstraction level of the spatiotemporal remote sensing data for applications. The collaborative transfer learning for data fusion of spatiotemporal data with land cover for different applications is investigated in [2]. The data fusion is trained in the collaborative nodes for land cover changes applications. However, collaborative schemes with resource constraints are unsuitable because they require large amounts of data for training drones.

The adaptive spatiotemporal data fusion-enabled generative adversarial network is suggested in [3]. The goal is to optimize the data fusion pattern and sequence for remote sensing data for drone applications. The anomaly detection in spatial and temporal traffic data patterns based on transfer and unsupervised learning for drone applications is discussed in [4]. Both known and unknown anomalies are identified for both data and services in the network. These studies [5], [6] extended the previous work and proposed the STOPPAGE and secure river spatiotemporal measuring for water calculation based on spatio temporal distribution, detection, and detection in distributed lands data. The land data is collected from remote sensing and placed at different levels with cloud computing. This work [7] suggested bean cultivation based on drone technology and satellite remote sensing data that consisted of stochastic spatial and temporal constraints for applications. However, the suggested CNN with the fixed gradient vanishes the scheme and needs more accuracy with nonlinear data.

These studies [8], [9], [10], [11], [12], [13], [16] presented applications for air pollution, parcel, traffic, pasture, forest crop, small object and soil in different trajectory locations, cropland change detection and time-based on remote sensory data from edge cloud satellites. Cropland change detection in remote sensing is hindered by spatial heterogeneity and temporal noise, leading to misaligned representations and erroneous change identification has been investigated. These studies suggested deep learning with different machine learning algorithms to train both supervised and unsupervised data to detect the optimal sequence pattern for data collection. The main objective of these studies was to collect images from different locations with their applications and design an accurate pattern for drones in the spatiotemporal spaces of given networks.

The flood, regional monitoring, flood risk, australian torrential event catelog and medicine delivery drone applications based on remote sensing data are presented in [14], [15], [17], [18]. The primary goal of these studies is to deliver medicine to flood-prone areas, torrential event catelog, determined flood risk in regional monitoring where image data is collected from satellites in real-time. All the studies presented different methods above. For instance, the remote sensing spatial-temporal

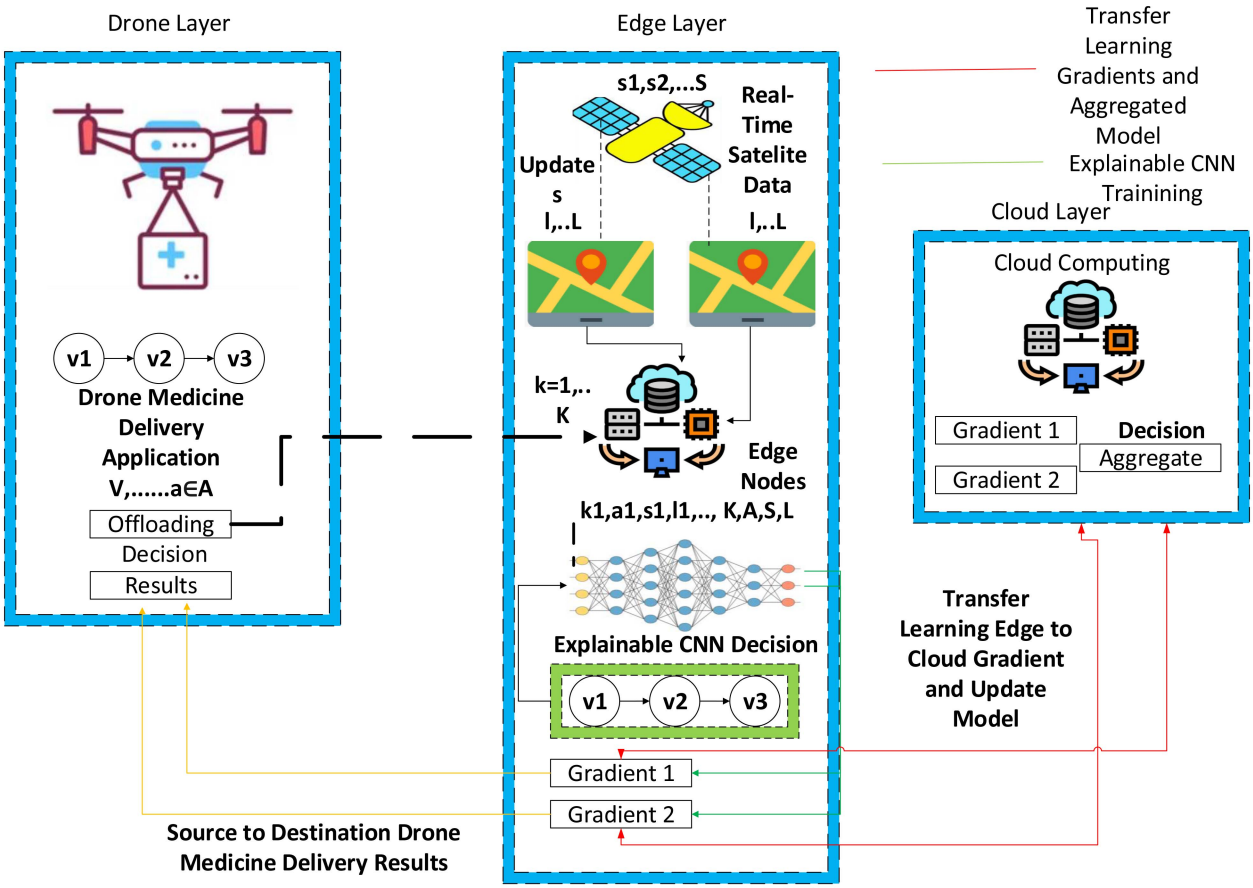


Fig. 1. Transfer learning-enabled system for drone medicine delivery based on SpatioTemporal remote sensing data in edge cloud networks.

scheme (RMSTS) for drone applications, the remote sensing drone delivery scheme (RMSDDS), and the remote sensing spatial-temporal transfer learning (RSSTTL). The primary objective was to achieve the highest score and deliver the medicine to the designated locations using the same information. However, these works should have considered the time factor more widely. These studies [19], [20] suggested that machine learning and reinforcement learning enabled remote sensing data collection and scheduling into different states to increase reward scores and improve the accuracy based on the score for UAV. Traditional light field (LF) image quality assessment (IQA) methods have primarily relied on full-reference (FR) or reduced-reference (RR) models, where pristine or partial reference information is required to measure distortions; however, such approaches, including PSNR, SSIM, and their LF-oriented extensions, are often impractical in real-world scenarios where reference images are unavailable. To overcome this limitation, blind image quality assessment (BIQA) techniques have been widely developed for natural 2D images, using either handcrafted statistical features (e.g., BRISQUE, NIQE, CORNIA) or deep learning-based feature extraction to capture perceptual distortions, though these methods are not directly applicable to LF data due to its additional angular and depth dimensions. Recent LF-IQA studies have attempted to incorporate spatial-angular consistency, disparity maps, and epipolar plane images (EPIs) to better model perceptual quality, but most existing approaches still depend on reference data or overlook the joint structural and perceptual characteristics of LF images. To address these challenges, Chai et al. [21] proposed a blind LF quality evaluator that leverages group-based representations and multiple plane-oriented perceptual features, effectively bridging the gap between conventional BIQA techniques and the unique requirements of LF image quality assessment.

To the best of our knowledge, the transfer learning and explainable CNN-enabled drone medicine delivery using spatiotemporal data from real-time satellite images in an edge cloud has not been studied yet. In this paper, we aim to improve medicine delivery by achieving a higher execution ratio and a higher success rate in the proposed system.

III. Proposed Plant Disease Detection System

We present the spatiotemporal framework comprising different processing mechanisms, e.g., application, management, and node, as illustrated in Fig. 1. We consider the many parcel delivery applications A that consist of different workflow tasks and have precedence constraints on workflows.

We consider three nodes: local drones, edge nodes, and cloud nodes. The local drone is directly connected to the edge nodes

TABLE II
SYMBOL NOTATIONS

Symbol	Description
A	All drone applications
V	Total number of tasks of drone application a
v	Particular drone application task
v_d	Data of drone task v
R	Collection of remote sensing data
s_r	Spatio of remote sensing data
T	Temporal data
t_r	Temporal of remote sensing data
v_r	Remote sensing for tasks
C	Collection of cloud computing
c	Particular cloud node
ϵ_c	Resource of cloud node
ζ_c	Speed of cloud node
K	Collection of Edge computing
k	Particular edge node
ϵ_k	Resource of edge node
ζ_k	speed of edge node
M	Collection of drone devices
m	distinct drone node
ϵ_m	Resource of drone node
ζ_m	Speed of drone node
$W[d, m, k, c]$	Transfer learning on different nodes
P	Total number of parameters
L	Total number of layers

for parcel delivery, allowing it to access real-time and updated satellite and spatial data. We demonstrate decision analysis at all nodes, including local drone applications that make offloading decisions and result analysis, edge nodes that make training decisions based on collected data, and the cloud layer that makes aggregation decisions using gradient training from edge cloud. We updated the satellite real-time values at the edge nodes for the drone application. The local edge nodes trained the pattern of different drone data at run-time based on convolutional neural networks and explained their decision at the softmax layer. The edge nodes trained the local model using a gradient, in which the source and destination patterns of the drones are trained with higher efficiency. The transfer learning is integrated between edge nodes and cloud computing for the gradient aggregation and sends back the aggregated model to the edge nodes. The optimal results are shared with the drone applications after the completion of the workflow process.

The preliminary task of the drone application is to offload parcel data from the source to the destination, where location data about the destination is collected from remote sensing nodes. Therefore, there are many scenarios of remote sensing data, such as data collection from satellites and processing them for location searching, parcel delivery, and surveillance. Therefore, for each workflow, we discuss the various scenarios based on remote sensing data in our framework. All workflow applications consist of dependent tasks, meaning each task depends on the previous tasks. In this framework, the process involves a sequential delivery of medicine through IoDT.

We consider the combinatorial problem; therefore, offloading and scheduling are the key methods. In offloading, all the parcels are initialized and delivered to the destinations. The offloading method is implemented at the drone application layer, where the decision is made whether data processing is to be offloaded. This decision concerns data sharing and training, where local drone devices have limitations due to resource constraints and cannot run and train on Big Data locally. We employed an explainable CNN architecture at the softmax layer to train the drone delivery data with satellite inputs. The explainability was ensured by integrating post-hoc interpretation methods to highlight how the model makes decisions. Specifically, we generated attention maps and feature activation visualizations (e.g., Grad-CAM and saliency maps) to identify the spatiotemporal features most relevant to drone path planning. These visualisations clearly demonstrate which critical regions in the satellite images influenced the CNN during the decision-making process for medicine delivery. For example, areas such as road networks, open fields, and no-fly zones were highlighted as dominant features.

A. Problem Formulation

Table II defines the mathematical notations of the problem formulation. We consider the A number of workflow applications in the proposed framework, e.g., $A = \{a = 1, \dots A\}$. Each workflow application consisted of tasks and tuples, e.g., $a = \{v, e, d, \dots, V, E, D\}$, where V is the workflow tasks, E is the number of communication points between tasks, and D is the original and shared data of the tasks. We collected spatiotemporal information from various sources, including location and destination-related data, as well as remote sensing data such as edges and clouds. We consider D multi-source data fusion in our framework, where d is the distinct data for the particular source for drone application. We consider different sources of information providers, e.g., S, where each piece of information is embedded in the cloud and edge nodes and represented by s, l, t, and d tuples. We defined the notations as follows: s is the source, l is the location, t is the time series temporal, and d is the data.

B. Mathematical Problem Formulation

Mathematically, we formulate the research problem as follows. Let V be the set of tasks, M local drone devices, K edge nodes, and R cloud nodes. Parameters: c_v, e_v, r_v are compute workloads (cycles) for local/edge/cloud phases of task v; u_v, d_v are upload/download bits; $\zeta_m, \zeta_k, \zeta_r$ are fixed service rates (cycles/s); $Up_{mk}, Up_{mr}, Down_{mk}, Down_{mr}$ are link capacities (bits/s); D_v is the deadline of v. Decision variables: $x_{vm}, y_{vk}, z_{vr} \in \{0, 1\}$ (assignment), with $x_{v\cdot} + y_{v\cdot} + z_{v\cdot} = 1$; auxiliary routing binaries w^{mk}_{vmk} and w^{mr}_{vmr} indicate an offload path $m \to k$ or $m \to r$ for task v; $o_v \in \{0, 1\}$ flags whether v offloads (edge or cloud). Scheduling variables $S^{\bullet}_v, C^{\bullet}_v$ represent start/finish times of each phase $\bullet \in \{\text{loc}, \text{up}, \text{rem}, \text{down}\}$; C_v is the final completion time. Big-M denotes a sufficiently large constant.

We determine the initial drone processing time to start tasks on the local drone devices in the following way:

$$\text{Local} = \sum_{v \in V} \sum_{m \in M} x_{vm} \frac{c_v}{\zeta_m}, \tag{1}$$

which determines the local processing time contribution across all drone tasks. Each x_{vm} selects the device m that executes v's local phase.

We determine the offloading time as follows.

$$\text{Offload} = \sum_{v\in V}\left[\sum_{m\in M}\sum_{k\in K} w_{vmk}^{mk}\left(\frac{u_v}{Up_{mk}}+\frac{d_v}{Down_{mk}}\right) + \sum_{m\in M}\sum_{r\in R} w_{vmr}^{mr}\left(\frac{u_v}{Up_{mr}}+\frac{d_v}{Down_{mr}}\right)\right] \quad (2)$$

which determines the upload/download time on the selected wireless links. The bilinear coupling between where the task was local (m) and where it executes remotely (k or r) is captured by the binaries w (linearized below).

We collect the edge sensing data from different edge nodes and execute it in the following way:

$$\text{Edge} = \sum_{v\in V}\sum_{k\in K} y_{vk}\frac{e_v}{\zeta_k}, \quad (3)$$

which determines remote processing at edge nodes.

We collect the remote sensing data from different cloud nodes and execute it in the following way:

$$\text{Remote} = \sum_{v\in V}\sum_{r\in R} z_{vr}\frac{r_v}{\zeta_r}, \quad (4)$$

which determines remote processing at cloud nodes.

We determined the per-task phase timing and precedence in the following way:

$$C_v^{\text{loc}} = S_v^{\text{loc}} + \sum_m x_{vm}\frac{c_v}{\zeta_m}, \quad (5)$$

$$C_v^{\text{up}} = S_v^{\text{up}} + \sum_{m,k} w_{vmk}^{mk}\frac{u_v}{Up_{mk}} + \sum_{m,r} w_{vmr}^{mr}\frac{u_v}{Up_{mr}}, \quad (6)$$

$$C_v^{\text{rem}} = S_v^{\text{rem}} + \sum_k y_{vk}\frac{e_v}{\zeta_k} + \sum_r z_{vr}\frac{r_v}{\zeta_r},\ S_v^{\text{down}} \geq C_v^{\text{rem}}, \quad (7)$$

$$C_v^{\text{down}} = S_v^{\text{down}} + \sum_{m,k} w_{vmk}^{mk}\frac{d_v}{Down_{mk}} + \sum_{m,r} w_{vmr}^{mr}\frac{d_v}{Down_{mr}}. \quad (8)$$

We determined the total time for all tasks in the following way (with local-only vs. offloaded completion gated by o_v):

$$o_v \geq \sum_k y_{vk}, \quad o_v \geq \sum_r z_{vr}, \quad o_v \in \{0,1\},$$
$$C_v \leq C_v^{\text{down}} + M(1-o_v), \quad C_v \leq C_v^{\text{loc}} + Mo_v,$$
$$\text{Time} = \sum_{v\in V} C_v, \quad C_v \geq 0. \quad (9)$$

We assume the score is the reward when all tasks are executed under their deadlines:

$$s_v \in \{0,1\}, \quad C_v \leq D_v + M(1-s_v), \quad \text{Score} = \sum_{v\in V} s_v, \quad (10)$$

which activates $s_v = 1$ only if task v completes by its deadline D_v.

We explicitly capture resource contention and non-overlap (combinatorial structure) in the following manner. For any two distinct tasks $v \neq w$ assigned to the same compute or link resource, one must precede the other:

$$\textbf{LocalCPU}(m):$$
$$S_v^{\text{loc}} \geq C_w^{\text{loc}}, \quad S_w^{\text{loc}} \geq C_v^{\text{loc}}, \quad b_{vwm}^L \in \{0,1\},$$
$$\textbf{EdgeCPU}(k):$$
$$S_v^{\text{rem}} \geq C_w^{\text{rem}}, \quad S_w^{\text{rem}} \geq C_v^{\text{rem}} - Mb_{vwk}^E, \quad b_{vwk}^E \in \{0,1\},$$
$$\textbf{CloudCPU}(r):$$
$$S_v^{\text{rem}} \geq C_w^{\text{rem}} \quad S_w^{\text{rem}} \geq C_v^{\text{rem}} \quad b_{vwr}^C \in \{0,1\},$$
$$\textbf{Uplink}(m{\to}k, r):$$
$$S_v^{\text{up}} \geq C_w^{\text{up}} - M(1-b_{vw\ell}^U), \quad S_w^{\text{up}} \geq C_v^{\text{up}} - Mb_{vw\ell}^U,$$
$$\textbf{Downlink}(m{\leftarrow}k, r):$$
$$S_v^{\text{down}} \geq C_w^{\text{down}} - M(1-b_{vw\ell}^D), \quad S_w^{\text{down}} \geq C_v^{\text{down}} - Mb_{vw\ell}^D. \quad (11)$$

where ℓ indexes a specific link; each pair of tasks sharing a resource gets a binary order variable b. These constraints ensure no overlaps and explicitly model contention. The routing binaries linearize the coupling between local and remote choices:

$$w_{vmk}^{mk} \leq x_{vm}, \quad w_{vmk}^{mk} \leq y_{vk}, \quad w_{vmk}^{mk} \geq x_{vm} + y_{vk} - 1$$
$$w_{vmr}^{mr} \leq x_{vm} \leq z_{vr}, \quad w_{vmr}^{mr} \geq x_{vm} + z_{vr} - 1, \quad \in \{0,1\}. \quad (12)$$

We consider the objective functions as a combinatorial problem that (i) maximizes rewarded completions and (ii) minimizes total completion time. The maximization is determined in the following way:

$$\max \sum_{v\in V} s_v, \quad (13)$$

which determines the maximum score during offloading and scheduling. We consider the minimization function as a convex (linear) objective on the continuous relaxation; we determined it in the following way (used either as Stage-2 given (13), or jointly via β-scalarization):

$$\min \sum_{v\in V} C_v, \max \beta \sum_v s_v - (1-\beta)\sum_v \frac{C_v}{\max_{v'} D_{v'}}, \in 0,1]. \quad (14)$$

which optimizes the total time of applications while preserving the primary goal of meeting deadlines.

a) Justification of the combinatorial structure and convex/concave aspects:

- The interdependencies arise from (a) precedence (8), (b) exclusive use of shared CPUs and links (11), and (c) coupled routing/placement (12). These yield integer (binary) decisions and disjunctive constraints, hence a mixed-integer linear program (MILP).

- The score objective $\sum s_v$ is linear (both convex and concave). Maximizing it is a concave optimization over the continuous relaxation, but integrality makes the overall problem combinatorial.
- The time objective $\sum C_v$ is linear; together with linear precedence and non-overlap (via big-M), the continuous relaxation is a convex polyhedron. Thus, the convex properties are exploited in bounding/relaxation, while integrality enforces combinatorial scheduling.
- The terms $\frac{c_v}{\zeta_m}, \frac{e_v}{\zeta_k}, \frac{r_v}{\zeta_r}$ and $\frac{u_v}{Up}, \frac{d_v}{Down}$ are linear because rates are fixed parameters. If one wishes to model bandwidth sharing or DVFS (rate as a decision), the resulting time = work/rate becomes convex; it can be handled via piecewise-linear convex approximations or SOCP, and plugged into (8) unchanged in spirit.

All symbols used above are now explicitly defined, overlapping tasks are prevented, link/CPU contention is enforced, and deadlines/scores are encoded with indicator binaries providing a rigorous optimisation framework consistent with the proposed formulation.

IV. Proposed OSPTS Algorithm-Based Methodology

This paper presents the OSPTS algorithm-based methodology, which is designed to optimize drone-assisted applications through the integration of three essential schemes: offloading, training, and scheduling. The methodology addresses the growing need for efficient computation and intelligent decision-making in drone ecosystems, particularly in domains such as medicine delivery, surveillance, environmental monitoring, and disaster management. To achieve this, we formulate the scheduling and offloading process as a combinatorial optimization problem, where the primary objective is to determine an optimal allocation of resources across heterogeneous nodes while meeting stringent quality of service requirements. The solution is obtained by leveraging advanced search techniques and convolutional neural networks (CNNs), which play a crucial role in analyzing the large-scale data collected by drones and in informing scheduling decisions with predictive intelligence. Furthermore, the OSPTS methodology is systematically divided into distinct phases, where each phase is carefully defined to capture a particular aspect of drone operations: the offloading phase ensures efficient utilization of edge, fog, and cloud nodes; the training phase incorporates transfer learning and model updates for accuracy improvements; and the scheduling phase applies knapsack-based optimization to balance profit, deadlines, and resource constraints. We define the OSPTS algorithm methodology with all schemes in the following subsections.

A. Drone Task Initiation and Offloading Scheme

The proposed Drone Task Initiation and Dynamic Offloading Scheme ensures that computationally intensive tasks generated by drones are executed efficiently through the dynamic selection of local, edge, or cloud execution. Initially, each drone task $v \in V$ is characterized by its computation demand θ_v, execution deadline δ_v, and data size s_v. The scheme first estimates the local execution time T_v^{local} for processing the task on the drone using (1). For potential offloading, the algorithm dynamically evaluates all available edge nodes $k \in K$ and cloud servers $c \in C$ by considering their respective communication bandwidths ζ, resource availability $\mathcal{R}$, and network latencies L. For each edge node, the offloading delay $T_{v,k}^{offload}$ is computed as the sum of the transmission time $\frac{s_v}{\zeta_k}$ and the computation time $\frac{\theta_v}{\mathcal{R}_{edge}(k)}$, while the overall cost is determined through a weighted function $Cost_{v,k} = \alpha L_{edge}(k) + \beta T_{v,k}^{offload}$, where α and β control the balance between latency sensitivity and computation efficiency. Similarly, for cloud servers, the offloading delay $T_{v,c}^{offload}$ and cost $Cost_{v,c}$ are calculated by incorporating ζ_c, $\mathcal{R}_{cloud}(c)$, and $L_{cloud}(c)$. The scheme dynamically compares these costs across all edge and cloud options, selects the node with the minimum cost, and updates its available resources after allocation. Finally, a local versus offloading trade-off is applied: if the local execution time T_v^{local} is lower than the minimum offloading cost, the task is executed directly on the drone; otherwise, it is offloaded to the best-selected edge or cloud node. In this way, the offloading scheme not only balances latency, bandwidth, and computational resources but also ensures that drones adaptively decide the most suitable execution environment for each task under dynamic network and resource conditions.

We design a dynamic offloading algorithm based on socket programming rules, where all local drones are connected to servers that act as edge nodes and cloud nodes located at different sites for distributed task execution. The proposed scheme enables drones to process customer, parcel, source, destination, and delivery data locally while establishing connections with remote edge and cloud servers for tasks that require higher computational power or global coordination. For example, in a medicine delivery scenario, the drone initially determines the local processing time T_v^{local} using (1) and evaluates whether offloading is beneficial by computing the transmission delay $\frac{s_v}{\zeta}$ and execution delay $\frac{\theta_v}{\mathcal{R}}$ at each candidate node. Unlike static schemes, the proposed dynamic offloading approach continuously evaluates multiple edge and cloud servers based on resource availability, network latency, and task deadlines. A cost function, $Cost = \alpha \times Latency + \beta \times OffloadingTime$, is used to compare all possible options, and the server (edge or cloud) with the minimum cost and sufficient resources is selected as the execution node. If no remote server can provide a better response time than local execution, the drone completes the task locally; otherwise, it offloads to the optimal edge or cloud server. In this way, the offloading decision $Offloading[v, m, d]$ is no longer limited to a binary wait-or-execute state. Instead, it represents an adaptive strategy that dynamically allocates tasks to the most efficient execution environment. After each offloading decision, server resources are updated to reflect current availability, ensuring that subsequent tasks are scheduled reasonably and efficiently. Thus, Algorithm 1 not only supports real-time socket-based connectivity between drones and servers but also minimizes overall latency, balances load across distributed resources, and ensures reliable and timely medicine delivery through adaptive task scheduling in Internet of Drone Things (IoDT) networks.

Algorithm 1: Drone Task Initiation and Dynamic Offloading Scheme (OSPTS).

Input: V (set of tasks), D (drones), C (cloud servers), K (edge nodes), M (available resources)
Output: Offloading decisions and execution results
Initialize resource availability: $\mathcal{R}_{edge}(k)$, $\mathcal{R}_{cloud}(c)$
Initialize latency parameters: $L_{edge}(k)$, $L_{cloud}(c)$
for each task $v \in V$ **do**
  Collect task requirements: computation demand θ_v, deadline δ_v, data size s_v
  Estimate local drone processing time T_v^{local} using (1)
    Initialize best_node ← null, min_cost ← ∞
  for each edge node $k \in K$ **do**
    Compute $T_{v,k}^{offload} = \frac{s_v}{\zeta_k} + \frac{\theta_v}{\mathcal{R}_{edge}(k)}$
    Compute overall cost: $Cost_{v,k} = \alpha L_{edge}(k) + \beta T_{v,k}^{offload}$
    if $Cost_{v,k} < min\_cost$ **and** $\mathcal{R}_{edge}(k)$ sufficient **then**
      best_node ← edge(k), min_cost ← $Cost_{v,k}$
    end if
  end for
  for each cloud server $c \in C$ **do**
    Compute $T_{v,c}^{offload} = \frac{s_v}{\zeta_c} + \frac{\theta_v}{\mathcal{R}_{cloud}(c)}$
    Compute overall cost: $Cost_{v,c} = \alpha L_{cloud}(c) + \beta T_{v,c}^{offload}$
    if $Cost_{v,c} < min\_cost$ **then**
      best_node ← cloud(c), min_cost ← $Cost_{v,c}$
    end if
  end for
  if $T_v^{local} < min\_cost$ **then**
    Execute task v locally on drone
  else
    Offload task v to best_node
    Update $\mathcal{R}_{edge/cloud}$(best_node) after allocation
  end if
end for
End

Algorithm 2: Transfer Learning Training, Aggregation, and Explainability Based on CNN Among Drone Edge Cloud.

Offloading[v, m, d]
Initialize pre-trained CNN model weights $W[, R, D, m, c, k]$
Load pre-trained CNN model at local devices
Establish a socket connection among nodes
Apply convolutional layers on sensing data and drone inputs
Initialize new weights W'
Load new dataset D, R for all tasks
Fine-tune the aggregated model on datasets D and R using backpropagation
Extract spatio-temporal features for decision-making
Apply Grad-CAM at the CNN softmax function to visualize critical regions influencing delivery
Generate saliency maps for feature importance validation
Share trained temporal and spatial dataset among nodes C, K, and M
Call knapsack scheduling Algorithm 3 with all components
Update weights W and W' using gradient descent
return Fine-tuned and explainable CNN model V, D, and R with visualization outputs

B. Transfer Learning Training and Aggregation Based on CNN Among Drone Edge Cloud

In our system, we collect remote sensing data related to lands, locations, and drone positions from different satellites attached to edge and cloud nodes. Therefore, remote sensing data is crucial for completing the goal of drone tasks in medicine delivery, from source to destination. We utilize various remote sensing data sources and employ a transfer learning strategy to connect all servers, enabling them to collect and train data for drone medicine delivery tasks.

For transfer learning, we connect all nodes using socket programming, ensuring that all nodes share the same operating system and interoperability to support the execution of task bytecode and compiled code. The decision node, added to the edge node, gathers trained and checked data from various nodes to make decisions about scheduling and predicting the best order for delivering medicines throughout the system. Transfer learning enabled nodes to train, test, and validate data based on an explainable CNN.

Algorithm 2 determines the training of spatial-temporal data on different nodes and combines them to form the aggregated node. The model trains temporal, spatial, and drone task data based on specific criteria, with each node applying CNN layers locally. Communication between nodes is maintained through socket protocols, as described in steps 1–5. After fine-tuning with backpropagation, we introduce explainability at the CNN softmax layer. Specifically, Grad-CAM is applied to highlight the critical spatio-temporal regions in satellite and drone data that influence decision-making (e.g., road networks, no-fly zones, and open fields). Saliency maps are also generated to validate feature importance and visually explain how the CNN prioritises input data during training. These visualisations ensure the interpretability of the model's decisions. The trained and explainable model is then shared among nodes, and the knapsack scheduling algorithm is applied to optimise scheduling scores and minimise processing time. Finally, all model weights are updated using gradient descent (steps 10–14). The algorithm thus provides both performance efficiency and transparency in drone delivery applications.

C. Knapsack Scheduling

We present the adaptive knapsack scheduling algorithm to schedule tasks across different nodes, aiming to complete all tasks as shown in Algorithm 3. In this algorithm, we explicitly formulate the scheduling of tasks as a knapsack-based optimization problem, where each task $v \in V$ must be assigned

to a computing node (m, k, c) under strict resource and timing constraints. For each task, we define the profit function $\text{profit}(v, m, k, c)$ as the accuracy gain or computational benefit achieved when the task is executed on node (m, k, c), particularly considering improvements obtained from transfer learning models trained on similar data, while the cost function $\text{cost}(v, m, k, c)$ represents the required consumption of heterogeneous resources such as CPU cycles, memory, bandwidth, and energy. The scheduling objective is therefore to maximize the overall gain-to-cost ratio, subject to the knapsack capacity constraints $\sum_{v \in V} \text{cost}(v, m, k, c) \leq \epsilon_{m,k,c}$ for all nodes, where $\epsilon_{m,k,c}$ denotes the available capacity of resources on each node. For every task, the algorithm evaluates whether it can be feasibly placed on a node without violating deadline constraints T_v and residual resource budgets. If so, it computes the combined scheduling score as $Score(v, m, k, c) = \frac{\text{profit}(v,m,k,c) + f_{TL}(D_v, model_{m,k,c})}{\text{cost}(v,m,k,c)}$, where f_{TL} denotes the accuracy or performance gain achieved through transfer learning on dataset D_v with the available pre-trained model $model_{m,k,c}$. The task is then mapped to the node that maximizes this score, and the resource capacity of the selected node is updated accordingly to reflect the consumed resources. If no feasible node exists for a task due to either capacity exhaustion or deadline violation, the task is offloaded to the cloud as a backup mechanism to ensure reliability. This integrated approach simultaneously considers task offloading, transfer learning benefits, and scheduling constraints, thereby transforming the resource allocation problem into a well-defined combinatorial knapsack optimization problem that explicitly accounts for deadlines, heterogeneous resource limits, and learning-based execution gains.

For scheduling, we consider a heterogeneous computing environment that includes drones, edge nodes, and cloud servers, and we sort out all available nodes based on their residual capacities and suitability for task execution. Each task is evaluated against these nodes to identify the best allocation option that meets the required quality of service (QoS) constraints during the delivery of medicine from source to destination. The primary objectives of this scheduling process are to maximize the overall scheduling score by considering both profit and transfer learning gain, to minimize the processing and response time of tasks, and to reduce the overall resource consumption during execution. In accordance with Algorithm 3, steps 1 to 10 correspond to the evaluation phase, where tasks are matched with candidate nodes based on the knapsack constraints, ensuring that capacity, deadline, and resource limitations are respected. The algorithm then iteratively assigns tasks to the most suitable nodes until the available task set is depleted. Steps 11 to 16 represent the termination phase, where unscheduled tasks ($v = 0$) are either offloaded to the cloud as a backup option or discarded if infeasible, thus guaranteeing that no pending tasks remain without consideration. This approach ensures efficient task placement across drones, edge, and cloud nodes, optimizing both performance and energy efficiency in critical medicine delivery applications.

Algorithm 3: Knapsack-Based Offloading, Transfer Learning, and Scheduling of Tasks.

Require: Task set $V = \{v_1, v_2, \ldots, v_n\}$, Node set C, K, M, resource capacities $\epsilon_{m,k,c}$, deadlines T_v, data D_v
Ensure: Optimal assignment of all tasks to computing nodes

1: **Define Knapsack Problem:**
2: For each task $v \in V$, define:

$$\text{profit}(v, m, k, c) = \text{accuracy transfer learning offloading}$$

$$\text{cost}(v, m, k, c) = \text{resource usage (CPU and more)}$$

3: Knapsack capacity constraint for each node (m, k, c):

$$\sum_{v \in V} cost(v, m, k, c) \leq \epsilon_{m,k,c}$$

4: **for** $v \in V$ **do**
5: Initialize $BestScore \leftarrow -\infty$, $BestNode \leftarrow \emptyset$
6: **for** $m \in M, k \in K, c \in C$ **do**
7: **if** $cost(v, m, k, c) \leq \epsilon_{m,k,c}$ **and** $deadline(v) \leq T_v$ **then**
8: Compute transfer learning gain:

$$gain(v, m, k, c) = f_{TL}(D_v, model_{m,k,c})$$

9: Compute scheduling score;
10: $Score(v, m, k, c) = \frac{profit(v,m,k,c) + gain(v,m,k,c)}{cost(v,m,k,c)}$;
11: **if** $Score(v, m, k, c) > BestScore$ **then**
12: $BestScore \leftarrow Score(v, m, k, c)$
13: $BestNode \leftarrow (m, k, c)$
14: **end if**
15: **end if**
16: **end for**
17: **if** $BestNode \neq \emptyset$ **then**
18: Assign task v to $BestNode$
19: Update resource capacity:

$$\epsilon_{BestNode} \leftarrow \epsilon_{BestNode} - cost(v, BestNode)$$

20: **else**
21: Offload task v to cloud (backup option)
22: **end if**
23: **end for**

V. Performance Evaluation

We integrated the proposed system, which is based on different programming languages, into the performance evaluation. We analyzed and defined the different datasets with different values. The simulation environment is configured based on different parameters such as languages, runtime, operating system, simulation time, repetition, and others, as shown in Table III.

There are many simulation parameters configured in the simulation. However, we mention only beneficial parameters in Table III. The simulation was implemented in Java, Python, and C on an x86 platform. We employed 20,000 drones to deliver 20,000 parcels to 200,000 customers using a client–server socket-based communication framework. Each drone was modeled as a flying UAV carrying up to 3 kg of medicine packages. The simulation ran for 5 hours, with each experiment repeated 10 times to ensure consistency and statistical reliability of the results. The most relevant simulation parameters are summarised

TABLE III
SIMULATION PARAMETERS

Notation	Description
Languages	JAVA, Python, C
Platform	X86
Drone Type	Flying Drone
Simulation time	5 hours
Repetition	10 times
Medicine Packages	3KG
Socket	Client Server
Drones	20000
Parcels	20,000
Customers	20,0000

in Table III, while other configurations (e.g., environmental factors, network variations) were kept constant to isolate the effect of our optimisation approach. These choices were made to strike a balance between computational cost and realistic drone delivery scenarios, enabling fair comparisons with existing baseline methods. By optimizing offloading time, scheduling time, and training accuracy, and validating performance through the delivery success rate, our approach achieves the highest score when parcels are delivered successfully from source to destination, demonstrating its superiority in both computational efficiency and practical applicability.

A. Spatial Temporal Datasets

In this paper, we consider and analyze different kinds of datasets, e.g., drone medicine delivery data, customer sender and receiver data, package data, and drone information. However, time and spatial location data are collected from different nodes, e.g., edges and point clouds, on various satellites. The point cloud collects the sender and receiver images used within the drone application for medicine delivery from the source to the destination.

Table IV presents the remote sensing datasets commonly used for drone-based applications, including their resolution, sample size, and preprocessing methods. Aerial and satellite imagery differ in temporal dynamics, with aerial images providing static high-resolution frames and satellite imagery offering dynamic coverage at daily or weekly intervals. LiDAR captures dense 3D point clouds, which undergo noise removal and down-sampling to improve efficiency. Hyperspectral and thermal datasets provide complementary spectral and temperature features, which are preprocessed through band selection, normalisation, and calibration. Radar-based data (SAR and GPR) contribute both surface and sub-surface information, filtered and aligned for clarity. To enable robust model training, multi-modal data are fused through spatial and temporal alignment. Imagery is geo-referenced with LiDAR point clouds, while hyperspectral and thermal features are co-registered using pixel-level alignment. Radar data are integrated via coordinate transformation, ensuring that features across modalities correspond to the exact spatial locations. This fusion provides a comprehensive dataset for drone-based sensing and decision-making.

Table V shows the data delivery dataset of drones from source to destination with different parameters. The drone number D is defined in the datasets. We exploited the 25 000 drone numbers with $V = 25\,000$ number of tasks in the dataset and implementation.

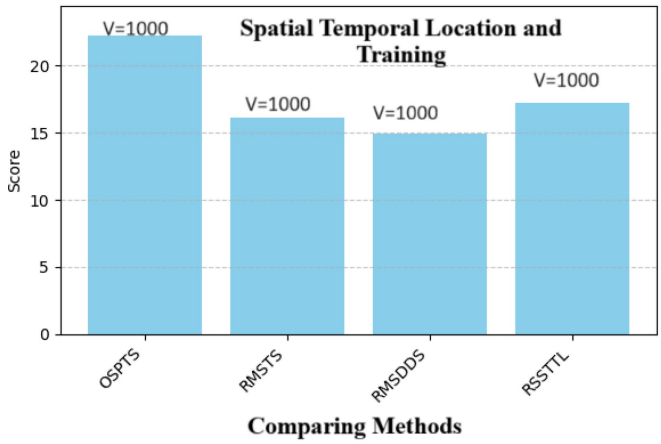


Fig. 2. Static offloading score medicine delivery based on point edge cloud.

Table VI presents the data package size, along with details of the customer, consumer, and receiver. It also shows the type of drone carrying the medicine, along with the original information needed to reach accurate locations while traveling through the network.

B. Result Analysis and Discussion

In this paper, we implement different baseline approaches as state-of-the-art comparison methods and propose schemes. For instance, RMSTS (remote sensing spatial-temporal scheme) [14] for drone data related to regional monitoring and applications, RMSDDS (remote sensing drone or different delivery scheme and monitoring schemes) flood risk on remote sensing data but application could be the drone and differents [15], and RSSTTL [17], [18] australian toreential event catalog and wheat growth monitoring. We assigned the different names to the baseline approaches and assumed them, they have used the data of remote sensing with the drone and applications.

Static offloading demonstrates that we schedule all tasks based on the available values of training data and computing nodes, and execute tasks within the given deadlines.

Fig. 2 shows the performance of local processing drone tasks related to medicine packages, sender, and receiver data without including real-time location and time in the network. We determined the local processing time based on the initial registration of the medicine within the local IoDT application, utilising socket programming, which recorded data related to the parcel, sender, and receiver, along with the provided information. Initially, we experimented with 2000 drone tasks for medicine delivery during simulation, e.g., $V = 2000$. We determined the score best based on the prediction sequence for meeting the deadline for task completion, and executed all tasks with a success ratio during execution. Fig. 2 shows that OSPTS scored higher than RMSTS, RMSDDS, and RSSTTL. The primary reason is that we trained the remote sensing data on different edge nodes and point clouds simultaneously to enhance

TABLE IV
TYPES OF REMOTE SENSING DATA FOR DRONE APPLICATIONS WITH RESOLUTION AND PREPROCESSING

Dataset Type	Example Modality	Spatial Resolution	Temporal Resolution	Sample Size	Preprocessing Steps
Imagery	Aerial Imagery Satellite Imagery	5–30 cm/pixel 0.3–10 m/pixel	Static Dynamic (daily/weekly)	10k+ images 50k+ images	Geo-referencing, normalization Cloud removal, resampling
LiDAR	3D Point Clouds	cm-level accuracy	Static	5M+ points	Noise removal, down-sampling
Hyperspectral	Spectral Signature	5–30 m/pixel, 200+ bands	Static	2k+ cubes	Band selection, normalization
Thermal	Temperature Distribution	10–50 cm/pixel	Static	8k+ frames	Calibration, denoising
Radar	Synthetic Aperture Radar (SAR) Ground Penetrating Radar (GPR)	1–10 m/pixel Sub-surface cm-level	Dynamic (hours–days) Static	20k+ images 1k+ profiles	Speckle filtering, normalization Signal filtering, alignment

TABLE V
DRONE LOCATIONS AND POINTS FOR MEDICINE DELIVERY

Drone	Location (X, Y)	Points (X, Y)
Drone 1	Source Destination	(10, 20) (50, 60)
Drone 2	Source Destination	(15, 25) (45, 55)
Drone 3	Source Destination	(20, 30) (40, 50)

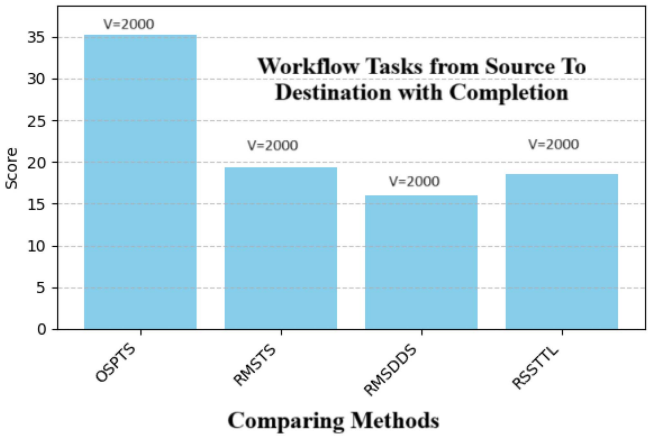


Fig. 3. Dynamic offloading score medicine delivery based on point edge cloud.

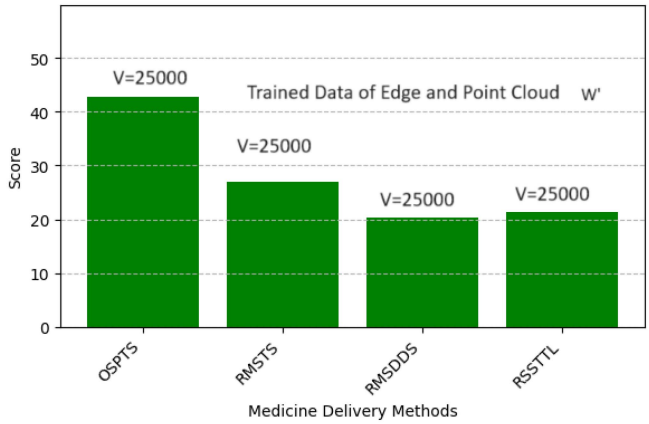


Fig. 4. Updated training remote sensing data and higher score medicine delivery based on point edge cloud.

the accuracy and score of drone medicine delivery, with optimal spatial sequence prediction from source to destination. Training different remote sensing on other nodes with more specifications is more optimal as compared to a point cloud. Existing studies have only trained data based on point cloud remote sensing data, resulting in fewer scores when tasks were not met, or the completion ratio was low during execution.

Offloading means that the drone application initiates the task of delivering medicine from a specific location to the destination. In dynamic offloading, remote sensing nodes, such as point cloud and fog nodes, collect data along different paths, allowing drones to navigate with less processing time and higher accuracy of completion without any harm. Fig. 3 shows that OSPTS scored higher than RMSTS, RMSDDS, and RSSTTL. The primary reason is that we trained the remote sensing data on different edge nodes and point clouds simultaneously to enhance the accuracy and score of drone medicine delivery, with optimal spatial sequence prediction from source to destination. The training on different cloud and edge nodes aggregated on the decision node, where dynamic data generated many good paths to get the optimal score for medicine delivery in networks.

Fig. 4 shows the runtime update of remote sensing data and scheduling during medicine delivery with the highest score prediction. The drone path is updated with the path of medicine delivery in the baseline model W. In contrast, the updated model W' has the optimal path and optimises the score more than the previous model. Fig. 4 shows that OSPTS scored higher than RMSTS, RMSDDS, and RSSTTL. The primary reason is that we trained the remote sensing data on different edge nodes and point clouds simultaneously to enhance the accuracy and score of drone medicine delivery, with optimal spatial sequence prediction from source to destination.

The time in Fig. 5 shows when the first processing starts for drone tasks, and trained data is downloaded from the decision node to find the best timing and sequence pattern for delivering medicine in the network with the highest score and the least amount of time spent on it. Fig. 5 shows the spatial-temporal training of data across different statuses, including initially loaded datasets and updated, dynamic datasets for medicine delivery from source to destination. The trained data has an optimal path, a sequence of source and destination patterns, and accurate information about locations across different geo-distributed areas. This helps drones navigate with a higher score ratio and less processing time, as shown in Fig. 5. Fig. 5 shows that OSPTS scored higher than RMSTS, RMSDDS, and RSSTTL. The main reason is that we trained the remote sensing data on different edge nodes and point clouds together to improve

TABLE VI
DRONE MEDICINE DELIVERY DETAILS

Customer	Consumer/Provider	Medicine Package Size	Drone Type	Source-Destination
John	Consumer	Small	Quadcopter	Source: (10, 20) Destination: (50, 60)
	Provider	Large	Octocopter	Source: (15, 25) Destination: (45, 55)
Alice	Consumer	Medium	Hexacopter	Source: (20, 30) Destination: (40, 50)

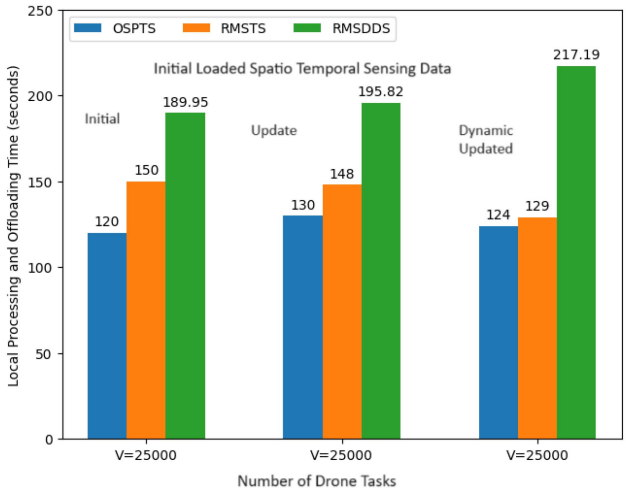


Fig. 5. Initial processing time and offloading time.

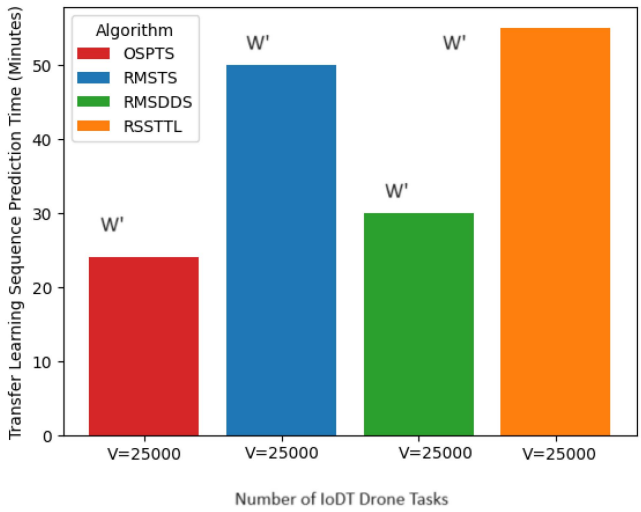


Fig. 7. Knapsack transfer learning scheduling for trained CNN-based IoDT task execution.

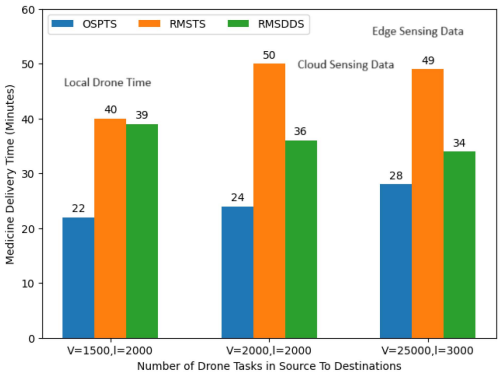


Fig. 6. Delivery time medicine delivery based on transfer learning point edge cloud.

processing time and the drone medicine delivery score, with optimal spatial sequence prediction from source to destination.

Fig. 6 establishes the initial processing duration, where drone assignments commence their execution, and processed data is downloaded from the decision node to determine the optimal temporal and sequential patterns for medicine delivery, aiming for the highest score and reduced execution time within the network. The spatial-temporal training depicted in Fig. 6 encompasses various states, locally trained data, and updated point and cloud datasets, tailored for medicine delivery from the source to the destination. The trained data encapsulates the optimal path, sequential pattern of source and destination, and precise location information across diverse geographically distributed areas, aiding drones in navigation with a superior score ratio and decreased processing time, as depicted in Fig. 6. Notably, Fig. 6 illustrates that OSPTS outperformed RMSTS, RMSDDS, and RSSTTL. This superiority stems from training on remote sensing data across different edge nodes and integrating point clouds, thereby enhancing processing time and the score of drone medicine delivery through optimal spatiotemporal sequence prediction from source to destination. The point cloud and edge data exploited transfer learning and shared their updated weights with a more accurate, optimal sequence pattern across different spatiotemporal data for drone applications.

We suggest knapsack scheduling, where all nodes execute the drone IoDT based on their resource availability, train the spatial-temporal data using a CNN, and share the results with the decision-aggregation node for final execution. Every node downloaded the trained model from the aggregated node for scheduling and executed all drone task jobs with the minimum processing time and highest score, as shown in Fig. 7. Whereas Fig. 7 demonstrates the superior performance of OSPTS over RMSTS, RMSDDS, and RSSTTL. This benefit comes from training remote sensing data across different edge nodes and combining point clouds. This reduces processing time and enhances the efficiency of drone medicine delivery by predicting the optimal spatial and temporal sequence from the source to the destination. Leveraging transfer learning, the point cloud and edge data collaboratively share their updated weights, resulting in a more accurate and optimal sequence pattern that incorporates various spatial-temporal data for drone applications.

VI. Conclusion and Future Work

This paper examines the use of drone technology for efficient medicine delivery across diverse locations by leveraging spatiotemporal remote sensing data collected from satellites, point cloud sources, and edge nodes. We introduced a Transfer Learning-enhanced spatiotemporal remote sensing training system integrated with the OSPTS algorithm to optimize medicine delivery tasks within the Internet of Drone Things framework. By modeling the problem as a combinatorial optimization challenge, the system utilizes search algorithms, CNNs, and transfer learning to improve data collection, task scheduling, and delivery efficiency. Simulation outcomes confirm that the proposed approach achieves higher delivery success rates, reduced processing times, and improved overall performance compared to existing methods. Despite these promising results, the system is subject to several limitations. Beyond the issues of service availability, resource costs, and security risks previously mentioned, there are additional constraints that must be acknowledged. Environmental factors such as adverse weather conditions (e.g., heavy rain, wind, or storms) can significantly impair drone performance and reliability. Moreover, the large-scale deployment of drone-based healthcare systems raises concerns about data privacy and confidentiality, particularly when handling sensitive medical information. Energy consumption, battery limitations, and air traffic management further represent operational challenges that were not fully addressed in this work.

Future work will focus on addressing these limitations by incorporating stronger security protocols and privacy-preserving mechanisms to protect sensitive medical and delivery data. To enhance operational resilience, future extensions will explore adaptive models that account for environmental variability and uncertain weather conditions. Additionally, we aim to integrate energy-efficient scheduling policies and battery optimization techniques to extend drone flight time and coverage. On the computational side, refining the data training process through federated learning and edge-cloud collaboration will further reduce latency and resource overhead. Lastly, developing standardized regulatory frameworks for large-scale drone deployment in healthcare logistics remains an open and vital area for future exploration. Overall, this work provides a foundation for advancing intelligent, secure, and resilient medicine delivery systems using drones, while highlighting critical areas where continued research is essential for real-world implementation.

Data Statements

We have utilised various types of datasets, including remote sensing, drones, point clouds, and edge data. The drone datasets can be found here: https://www.kaggle.com/datasets/dasmehdixtr/drone-dataset-uav and https://www.esri.com/en-us/arcgis/products/arcgis-reality/resources/sample-drone-datasets.

References

[1] A. Ferchichi, A. B. Abbes, V. Barra, and I. R. Farah, "Forecasting vegetation indices from spatio-temporal remotely sensed data using deep learning-based approaches: A systematic literature review," *Ecol. Informat.*, vol. 68, 2022, Art. no. 101552.

[2] X. Meng, Q. Liu, F. Shao, and S. Li, "Spatio–temporal–spectral collaborative learning for spatio–temporal fusion with land cover changes," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5704116.

[3] Q. Liu, X. Meng, F. Shao, and S. Li, "PSTAF-GAN: Progressive spatio-temporal attention fusion method based on generative adversarial network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5408513.

[4] T. M. Tran, D. C. Bui, T. V. Nguyen, and K. Nguyen, "Transformer-based spatio-temporal unsupervised traffic anomaly detection in aerial videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 9, pp. 8292–8309, Sep. 2024.

[5] K. S. Rautela, S. Singh, and M. K. Goyal, "Characterizing the spatio-temporal distribution, detection, and prediction of aerosol atmospheric rivers on a global scale," *J. Environ. Manage.*, vol. 351, 2024, Art. no. 119675.

[6] S. Ghosh, A. Mukherjee, S. K. Ghosh, and R. Buyya, "STOPPAGE: Spatio-temporal data-driven cloud-fog-edge computing framework for pandemic monitoring and management," *Softw.: Pract. Exp.*, vol. 52, no. 12, pp. 2700–2726, 2022.

[7] N. Devi, K. K. Sarma, and S. Laskar, "Design of an intelligent bean cultivation approach using computer vision, IoT and spatio-temporal deep learning structures," *Ecol. Informat.*, vol. 75, 2023, Art. no. 102044.

[8] A. C. de Araujo and A. Etemad, "End-to-end prediction of parcel delivery time with deep learning for smart-city applications," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 17043–17056, Dec. 2021.

[9] N. Awan et al., "Modeling dynamic spatio-temporal correlations for urban traffic flows prediction," *IEEE Access*, vol. 9, pp. 26502–26511, 2021.

[10] M. Rangwala et al., "DeepPaSTL: Spatio-temporal deep learning methods for predicting long-term pasture terrains using synthetic datasets," *Agronomy*, vol. 11, no. 11, 2021, Art. no. 2245.

[11] A. Qayum, F. Ahmad, R. Arya, and R. K. Singh, "Predictive modeling of forest fire using geospatial tools and strategic allocation of resources: Eforestfire," *Stochastic Environ. Res. Risk Assessment*, vol. 34, no. 12, pp. 2259–2275, 2020.

[12] M. Van Lier, M. Van Leeuwen, B. Van Manen, L. Kampmeijer, and N. Boehrer, "Evaluation of spatio-temporal small object detection in real-world adverse weather conditions," in *Proc. Winter Conf. Appl. Comput. Vis.*, 2025, pp. 844–855.

[13] J. Lv et al., "STMTNet: Spatio-Temporal Multiscale Triad Network for Cropland Change Detection in Remote Sensing Images," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 18, pp. 26005–26020, Sep. 2025.

[14] R. J. McCarroll, D. M. Kennedy, J. Liu, B. Allan, and D. Ierodiaconou, "Design and application of coastal erosion indicators using satellite and drone data for a regional monitoring program," *Ocean Coastal Manage.*, vol. 253, 2024, Art. no. 107146.

[15] D. Wang, W. Shao, J. Liu, H. Su, G. Zhang, and X. Fu, "Exploring spatio-temporal dynamics of future extreme precipitation, runoff, and flood risk in the hanjiang river basin, China," *Remote Sens.*, vol. 16, no. 21, 2024, Art. no. 3980.

[16] Z. Wei, S. Wang, J. Peng, Y. Li, L. Yue, and Y. Li, "An algorithm for spatio-temporal trajectory conflict risk identification in intersections considering lateral vehicle movement," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 9, pp. 13585–13602, Sep. 2025.

[17] M. Heiser, J. Hübl, and C. Scheidl, "Completeness analyses of the austrian torrential event catalog," *Landslides*, vol. 16, no. 11, pp. 2115–2126, 2019.

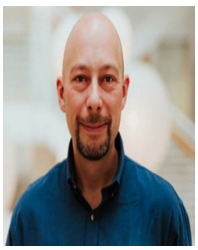
[18] Y. Li et al., "A spatio-temporal fusion framework of UAV and satellite imagery for winter wheat growth monitoring," *Drones*, vol. 7, no. 1, 2022, Art. no. 23.

[19] Y. Ding, Z. Yang, Q.-V. Pham, Y. Hu, Z. Zhang, and M. Shikh-Bahaei, "Distributed machine learning for UAV swarms: Computing, sensing, and semantics," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 7447–7473, Mar. 2024.

[20] S. Abbes and S. Rekhis, "Reinforcement learning for intelligent sensor virtualization and provisioning in Internet of Vehicles (IoV)," *IEEE Access*, vol. 12, pp. 54352–54370, 2024.

[21] X. Chai, F. Shao, Q. Jiang, X. Wang, L. Xu, and Y.-S. Ho, "Blind quality evaluator of light field images by group-based representations and multiple plane-oriented perceptual characteristics," *IEEE Trans. Multimedia*, vol. 26, pp. 607–622, 2023.

Abdullah Lakhan received the PhD degree in computer science and technology. He was a postdoctoral researcher with the Kristiania University of Applied Science. He is currently an associate professor with the Department of Cybersecurity, Dawood University of Engineering and Technology, Pakistan. He has authored or coauthored more than 100 high-quality Level 1 and Level 2 conference papers and journal articles. He has been a reviewer and a guest editor for Level 1 and Level 2 journals. He has co-organized several international conferences and regularly serves as a peer reviewer and committee member for international conferences and journals.



Tor-Morten Grønli (Member, IEEE) received the Master of Technology degree with distinction from Brunel University, London, U.K., in 2007, and the PhD degree in computer science from the College of Engineering Design and Physical Sciences, Department of Computer Science, Brunel University, London, U.K., in 2011. He is currently a professor with the Department of Technology, Kristiania University College, Norway. He is a visiting research scholar with the Department of Information Technology Management, Copenhagen Business School and joint affiliate of the Center of Business Data Analytics. He is the founder and director of the Mobile Technology Lab, Kristiania University College. He has coauthored more than 70 papers in books, conferences, and journals. His research interests include context awareness, mobile and pervasive computing, and the Internet of Things. He is a co-organizer of several International Conferences on Mobile Web and general chair for Norwegian Conference on ICT. He serves as TPC and regularly reviews for AINA, CompSac, SAC, HICSS, IEEE Big Data, COMPSAC, WoTBD, IEEE-WiMob, Percom, and ICTC. He has also guest edited various special issues in international journals, such as Journal of Computers & Electrical Engineering, and Journal of Future Generation Computer Systems, . He is a reviewer for several high-ranking journals and is on the editorial board of International Journal of Pervasive Computing and Communications, Journal of Online Information Review, and Computers & Electrical Engineering.

Ahmet Soylu received the PhD degree in computer science from the University of Leuven in 2012. He was a senior scientist with SINTEF Digital, senior engineer with DNV, postdoctoral research fellow with University of Oslo (UiO), and visiting researcher with the University of Oxford. He is currently a full professor of computer science and head of the group for Innovation, Digital Transformation, and Sustainability, OsloMet – Oslo Metropolitan University, and has adjunct positions with (UiO) and Norwegian University of Science and Technology. His main research interests include ontology-driven information systems and ontology-driven information systems design from a human-computer interaction perspective. His current research interests include data integration, management, and access to industrial and open data using semantic web and linked data technologies and principles. His research interests include end-user development, the Semantic Web, knowledge representation, digitalization, and Big Data.

Ghulam Muhammad (Senior Member, IEEE) received the BS degree in computer science and engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 1997, the MS degree in knowledge-based information engineering in 2003, and the PhD degree in electrical and computer engineering from Toyohashi University and Technology, Toyohashi, Japan, in 2006. He is currently a professor with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He holds three U.S. patents and has authored or coauthored more than 300 publications, including IEEE/ACM/Springer/Elsevier journals, and flagship conference papers. His research interests include AI, machine learning, image and speech processing, and smart healthcare. Prof. Muhammad was the recipient of the Japan Society for Promotion and Science fellowship from the Ministry of Education, Culture, Sports, Science, and Technology, Japan.

Qurat-ul-ain Mastoi (Member IEEE) received the PhD degree from the University of Malaya, where she focused on interdisciplinary research that bridges the gap between technology and its applications in crucial sectors. She is currently a dedicated academic, researcher, and a permanent faculty member with the University of the West of England Bristol, U.K. Her research interest include cybersecurity, healthcare technology, agricultural technology, and IoT security, where she is committed to advancing knowledge and innovation. Her work has contributed to the development of sustainable solutions aimed at improving human health and agricultural productivity. In recognition of her outstanding work, the Royal Academy of Engineering in the U.K. awarded her the prestigious Global Talent Award under the Exceptional Talent program. This honor highlights her significant impact in the fields of AI in healthcare, IoT security and agricultural technology, reinforcing her status as a leading expert and innovator in her field.

Huaming Wu (Senior Member, IEEE) received the BE and MS degrees in electrical engineering from the Harbin Institute of Technology, China in 2009 and 2011, respectively, and the PhD (highest Hons.) degree in computer science, Freie Universität Berlin, Germany, in 2015. He is currently a professor with the Center for Applied Mathematics, Tianjin University, China. His research interests include mobile cloud computing, edge computing, Internet of Things, deep learning, complex networks, and DNA storage.