



# Digital-twins and machine learning-assisted stable, energy-aware unmanned aerial and ground vehicles delivery in blockchain-enabled crowdsourcing framework

Feruz Elmay<sup>a,b</sup>, Maha Kadadha<sup>b</sup>, Shakti Singh<sup>a,b</sup>, Rabeb Mizouni<sup>a,b,\*</sup>, Hadi Otrok<sup>a,b</sup>, Azzam Mourad<sup>b,c</sup>

<sup>a</sup> Center for Cyber-Physical Systems (C2PS), Khalifa University, Abu Dhabi, United Arab Emirates

<sup>b</sup> Computer Science Department, Khalifa University, Abu Dhabi, United Arab Emirates

<sup>c</sup> Artificial Intelligence & Cyber Systems Research Center, Department of CSM, Lebanese American University, Beirut, Lebanon

## ARTICLE INFO

### Keywords:

Blockchain  
Crowdsourcing  
Digital twins  
Last-mile delivery  
Machine learning  
Smart contract

## ABSTRACT

Rising demand in last-mile delivery has heightened sustainability, cost, and delay concerns in traditional logistics. Unmanned Aerial Vehicles (UAVs) offer a transformative solution to ground vehicles due to their efficiency, low emission, and ability to avoid traffic. However, realizing their potential in crowdsourced delivery faces fundamental barriers: limited flight time due to highly unpredictable energy consumption that varies significantly with flight environment and payload. Existing solutions, such as simplified payload-based energy models, in-transit recharging, or pairing UAVs with trucks, often overlook the effect of flight environment uncertainty and fluctuating energy demands. These approaches are typically tailored for structured delivery networks, making them unsuitable for crowdsourced systems. Moreover, the absence of transparent, accountable frameworks severely limits adoption for high-value perishable deliveries and platform scalability. This paper presents a comprehensive energy-aware, secure, and transparent framework that fundamentally addresses these industry-critical challenges through the integration of UAVs and ground vehicles within a crowdsourced last-mile delivery system. It features a Random Forest-based energy consumption model that considers the UAV's and flight environment data, as well as its payload. Additionally, it employs a delivery-success probability aware Gale-Shapely game-based task allocation mechanism to maximize Quality of Service (QoS) and Quality of Delivery (QoD). Digital twins are proposed and modeled for transparent and real-time package storage and status monitoring. Experiments show that the trained energy consumption model achieves a mean absolute error (MAE) of 1.66. The allocation evaluation results highlight that the proposed system improves the QoS by at least 37%, delivery success by at least 45%, worker reputation and payoff by at least 32% and 22%, respectively, compared to the benchmarks.

## 1. Introduction

Supply-chain logistics involves moving goods from the origin to the final customer, including production, transportation, and warehousing. A critical yet often costly phase within this system is the last-mile delivery (LMD), the final step where goods are transported from a distribution center or local hub to the end customer [1]. Fueled by rapid growth in e-commerce and online orders, the global LMD market was valued at \$108.1 billion in 2020 and is forecast to reach \$200 billion by 2027 [2]. Rising demand has driven a sharp increase in delivery vehicle mileage, exacerbating urban congestion and carbon emissions; city logistics alone contribute a sizeable share of the 70% of global

emissions attributed to urban areas [3]. At the same time, industry data show that LMD's share of end-to-end shipping costs climbed from 41% in 2018 to over 50% in 2023 [4], underscoring the economic urgency for more efficient, lower-carbon delivery solutions.

Crowdsourced delivery has emerged as a promising response to these pressures by turning independent drivers into on-demand workforce. Studies show that this approach can shorten delivery times and lower operating costs for retailers and platforms alike [5–7]. Most systems rank candidate workers with a Quality of Service (QoS) metric that considers factors such as the workers' historical reputation,

\* Corresponding author. Khalifa University of Science and Technology, P.O. Box 127788, Abu Dhabi, United Arab Emirates

E-mail addresses: [feruz.elmay@ku.ac.ae](mailto:feruz.elmay@ku.ac.ae) (F. Elmay), [maha.kadadha@ku.ac.ae](mailto:maha.kadadha@ku.ac.ae) (M. Kadadha), [shakti.singh@ku.ac.ae](mailto:shakti.singh@ku.ac.ae) (S. Singh), [rabeb.mizouni@ku.ac.ae](mailto:rabeb.mizouni@ku.ac.ae) (R. Mizouni), [hadi.otrok@ku.ac.ae](mailto:hadi.otrok@ku.ac.ae) (H. Otrok), [azzam.mourad@ku.ac.ae](mailto:azzam.mourad@ku.ac.ae) (A. Mourad).

<https://doi.org/10.1016/j.future.2025.108004>

Received 6 February 2025; Received in revised form 19 June 2025; Accepted 26 June 2025

Available online 8 July 2025

0167-739X/© 2025 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

proximity to the task location, and estimated flight endurance, when Unmanned Aerial Vehicles (UAVs) are involved [7–9].

A significant drawback of these approaches is the lack of guaranteed task acceptance, as tasks are allocated to a predefined group of available workers under the assumption that each worker will accept the assigned task. This assumption increases the likelihood of canceled assignments, which can unfairly penalize the worker's reputation, lead to task expiry without completion, or even result in potential package damage. To improve reliability, several studies such as [6,10–13] integrate worker and task preferences in the allocation mechanism, while studies such as [14,15] embed machine-learning (ML) models that predict worker behavior or delivery success probabilities before assignment.

As crowdsourced delivery gains traction, the secure and timely delivery of packages; especially those that are sensitive and perishable, necessitates a well-coordinated system that effectively balances delivery speed with service quality. Among the capable solutions, *Unmanned Aerial Vehicles (UAVs)* as well as *Digital Twins* are one of the most researched and promising solutions for time- and cost-efficient package delivery [6,14,16,17]. UAVs offer compelling advantages for LMD due to their lighter weight, traffic-independent navigation, and environmental efficiency compared to ground vehicles (GVs). UAVs are expected to be a crucial part of several industries and sectors, some of which are already in operation: Wing, a delivery service, has been responsible for more than 350,000 deliveries across three continents in cooperation with retail companies [18]. Moreover, leading logistics enterprises including Amazon Prime Air, DHL ParcelCopter, and UPS Flight Forward are actively exploring the UAV delivery services [6,19,20].

In crowdsourcing contexts, UAVs enable companies to efficiently scale for fluctuating demands and increase coverage without maintaining dedicated fleets as highlighted in [6,9,16,21,22]. A critical obstacle in employing UAV delivery, however, is the restricted flight time due to battery capacity constraints. This challenge is compounded by fluctuating energy consumption patterns highly influenced by flight environment and payload variables.

Previous approaches have proposed to estimate the flight time through computations prior to delivery assignments [6,23], replenish the energy during delivery [9], or couple UAVs with delivery trucks to conserve energy [24]. Despite these advances, significant research gaps remain unaddressed in the integration of UAVs within crowdsourced LMD frameworks. Existing energy consumption models predominantly rely on simplified assumptions or are calibrated for controlled environments, failing to account for the dynamic and unpredictable conditions inherent in crowdsourced operations. Without accurate consumption estimates, platforms cannot guarantee that a UAV assigned to a task will complete it.

In parallel, current crowdsourced delivery platforms suffer from insufficient transparency and trust mechanisms, deterring requesters from utilizing the service for high-value goods, and quality workers from participating due to concerns about fair evaluation. Task-allocation methods further compound the issue; many optimize platform throughput at the expense of worker preferences, or they lack the stability properties required for reliable service under fluctuating supply and demand. Finally, conventional tracking services report little beyond geolocation, leaving package condition unmonitored and accountability gaps unresolved. Together, these limitations restrict the practical deployment of hybrid UAV–GV crowdsourced delivery networks.

*Digital twins* (DTs) technology presents a promising solution to transparency and trust challenges. DTs represent virtual counterparts of physical objects or systems, allowing for detailed simulations that closely mimic the behavior of their physical counterparts [25]. In supply chains, DTs enable transparent process management, status simulation, and enhanced decision support [26–30]. For crowdsourced delivery, they facilitate transparent package condition monitoring and objective worker evaluation based on delivery quality rather than

mere completion status. When integrated with blockchain technology, DTs create an immutable record of package handling that strengthens accountability throughout the delivery process.

This paper presents an energy-efficient, blockchain-based framework for integrating autonomous UAVs and GV in crowdsourced LMD. It employs prediction-based Gale-shapley matching to ensure the stability of the assignments by considering the preferences of both the workers and requesters. It is hosted on the Ethereum blockchain to enable a secure and automated allocation and ensure platform data security and participants' accountability. The matching algorithm incorporates estimated delivery success probability for both the GV and UAVs. In the case of UAVs, we propose a machine learning-based energy estimation model to forecast energy requirements accurately, factoring in task payload and flight environment. Similarly, the estimation model for GV employs task and worker characteristics and environmental factors to predict the probability of successful delivery. In the post-allocation stage, the framework employs package DTs for real-time status updates and worker performance assessment based on actual delivery outcomes.

In essence, the main contribution of this work is a framework employing hybrid crowdsourced workers for LMD that:

- Introduces a Random Forest-based predictive model for UAV energy consumption that precisely quantifies the impact of flight environment parameters, device operational conditions, and payload variations, enabling energy-aware task allocation,
- Implements worker-specific machine learning models that predict delivery success probability by analyzing multidimensional factors including environmental conditions, worker behavioral patterns, and task-specific attributes, to minimize the number of canceled and damaged packages,
- Designs and implements a blockchain-based stable allocation mechanism using Gale-Shapley matching that optimizes task distribution through quantifiable Quality of Service (QoS) and Quality of Request (QoR) metrics,
- Develops an integrated digital twin framework for continuous real-time package monitoring, status simulation based on environmental conditions, violations alerts generation, data-driven worker evaluation for improved reputation assessment accuracy.

The remainder of this paper is structured as follows. Section 2 reviews the background literature and presents the related works. Section 3 presents the proposed framework and the enabling components. Section 4 discusses the performance evaluation of the proposed system in comparison to benchmarks. Finally, Section 5 concludes the paper and suggests future works.

## 2. Related work

### 2.1. Crowdsourced UAVs: Last-mile delivery and energy consumption models

In [23], the authors present a payload-driven energy model using the Battery Consumption Rate (BCR). The authors conduct experimental flights and employ linear regression to model the relationship between the battery state-of-charge and payload weights. The authors of [31] approach energy consumption as a piecewise linear function of flight time, considering atmospheric events, specifically wind and rain. The problem is framed as a vehicle routing problem and solved using Benders' combinatorial cuts. In [32], the authors conduct an empirical study to model the battery performance of drones in various flight scenarios and present a black-box regression model to estimate the energy consumption of the drones. The model takes a total of nine parameters, including the drone motion, total weight, wind speed, and altitude. The model, however, is essentially summarized as a function of payload and speed [33].

Alternatively, the work discussed in [34] presents an LSTM-based architecture to estimate the energy consumption of drones. The authors use two bidirectional LSTM models stacked together and trained using the dataset found in [35], on a total of 21 features. Utilizing the same data, the authors of [36] propose a deep learning-based energy model using Temporal Convolutional Network (TCNs) as the core architecture. The model processes the time varying features such as velocity through the TCN and concatenates the time-invariant features through a dense layer. The authors of [37] compare the performance of two mathematical models and a Random Forest model in predicting the energy consumption of UAVs. The authors use the mathematical models to analyze how parameters such as drone mass, payload, rotor airflow and external conditions such as altitude and airspeed affect the power use, while a Random Forest Regressor is applied to compare predicted energy consumption with real flight data.

In the context of crowdsourcing, several studies tackle the energy constraint of UAVs by coupling them with ground vehicles. The work presented in [22] explores an energy conserving strategy in which UAVs are launched from and return to public vehicles during delivery. The task allocation is optimized using a MILP model that enforces each UAVs' maximum flight range. Alternatively, the authors of [21] tackle UAV energy limitations by proposing a delivery scheme in which energy-constrained UAVs can recharge by landing on social buses, thereby extending their operational range. The authors structure the problem as a multi-objective optimization to determine efficient delivery schedules, allowing the UAVs to dynamically align their flight path with bus routes.

The authors of [9] propose a cooperative delivery framework involving UAVs and crowdsourced taxis to maximize coverage, modeling energy consumption as a constant rate dependent on payload. Similarly, in [24], the authors present a truck-drone collaboration for delivery in which UAVs are launched from trucks to deliver packages. The work adopts the consumption model discussed in [38] to estimate the flight time and flight speed of UAVs based on the combined weight of UAV and payload, lift ratio, and UAV-specific energy-loss parameters, however do not consider the influence of weather conditions on the energy consumption.

While these aforementioned works contribute greatly to the literature, they exhibit certain limitations, as summarized in Table 1. Works such as [9,23,36] focus on limited features, neglect environmental uncertainty, and are tailored to specific UAVs and controlled delivery settings. As a result, they fail to capture the variability of in-flight energy consumption across diverse flight environments and crowdsourced delivery systems. Moreover, existing research integrating UAVs in crowdsourced delivery frameworks [6,9,21,22] overlooks critical aspects such as package security during transit and trust mechanisms between requesters and workers. Our study addresses these gaps by proposing a comprehensive prediction-based UAV energy model that considers device, task, and flight environment variables. Additionally, DTs and blockchain technology are proposed to enhance package security and build trust among stakeholders in the crowdsourced setting.

## 2.2. Crowdsourced vehicles delivery: Centralized and blockchain-based platforms

Several studies aim to boost delivery success by selecting high-quality workers based on computed metrics such as Quality of Service (QoS), that usually considers factors such as the workers' historical reputation, proximity to the task location, and confidence [6,7,16]. Alternatively, other studies such as [8,39–42] leverage data analysis and machine learning models to improve task success rates. For instance, the authors of [8] use Random Forests to predict the shipment request status by estimating the bidding, acceptance, and delivery probabilities.

In [39], the authors present agent-based simulations to forecast the on-time delivery of packages in crowd-shipping settings and dynamically reallocate tasks based on predicted delays. The authors of [40]

estimate the success of crowdsourced software development by analyzing task-specific requirements, while the authors of [41,42] focus on predicting the completion likelihood of sensing tasks considering the mobility patterns of workers. However, these approaches uniformly assume that available workers will agree to the assignment, a presumption that can leave tasks unfulfilled, and in turn compromise both reliability and fairness of crowdsourcing systems.

To account for the challenges posed by such assumptions, studies such as [10–13] have incorporated preference-based matching and predictive profiling into their allocation mechanisms. Preference-based methods allow the workers and tasks to express mutual interests. For instance, the authors of [10] consider workers' temporal preferences, while the authors of [13] construct the preferences of tasks based on worker's arrival times. In [12], the authors evaluate recommendation scores by considering worker reliability and preferences, while the authors of [11] identify the top  $k$  workers best suited for a task and top  $k$  tasks best suited for a worker based on category similarity scoring combined with worker historical performance.

Similarly, predictive models are used to forecast a worker's willingness to accept a task. The works discussed in [15,43] present random forest-based recruitment models, where task acceptance is predicted based on the worker and environment features. Similarly, the authors of [1] use logistic regression to forecast the willingness of in-store crowdsourced agents to undertake delivery tasks. In [44], the authors present a two-echelon crowdshipping network that integrates both worker acceptance probabilities and customer willingness to use the service, while the work discussed in [14] considers worker behavior, task attributes, and environmental factors to forecast delivery success in dynamic crowdsourced networks.

While recent models have advanced the reliability and fairness of task assignments, centralized crowdsourcing platforms still face broader challenges related to data integrity and trust, which can negatively impact stakeholder engagement and satisfaction. To mitigate these concerns, blockchain technology is increasingly adopted to provide secure, reliable, and trustworthy crowdsourcing platforms [14, 16,45–48]. Blockchain is a decentralized, secure, and immutable ledger maintained by a peer-to-peer network of nodes that validate transactions using consensus mechanisms such as Proof of Work (PoW), Proof of Stake (PoS), and Proof of Authority (PoA) [49–51].

The authors of [52] investigate how blockchain integration with AI and IoT technologies can enhance transparency, reliability, and traceability in delivery operations, while the authors of [45] explore the use of blockchain to facilitate secure data exchanges among stakeholders. Additionally, various studies, including [14,16,46–48], utilize blockchain for secure task distribution and data management in LMD systems. These studies employ different task allocation strategies, such as Gale-Shapley matching [16], offer-order based method [46], and Greedy algorithms [14,47], all implemented within smart contracts to ensure secure and autonomous operations.

Despite the efforts to improve task allocation in crowdsourced systems, several key limitations persist. First, approaches such as [10–12] attempt to integrate preferences in the allocation process but typically consider only one-sided preferences. Second, while works such as [13, 44] consider both-sided preferences, preferences are either computed based category similarity or are generated between predefined ranges. Third, majority of the works such as [16,41,42,46–48] evaluate delivery outcomes in binary terms, completed or not, without factoring in the quality of delivery. To address these gaps, our study presents two-sided preferences and delivery success probability-aware allocation mechanism. In addition, DT-enabled delivery feedback is used to evaluate worker performance beyond binary task completion.

## 2.3. Digital twins in supply chain and crowdsourced delivery

Due to their advantages, DTs have been increasingly adopted in supply chains for process management, real-time status simulation, and

**Table 1**  
Comparison of existing related works.

Reference	UAV energy models			Crowdsourcing frameworks					
	Approach	Considered factors	Dataset	Vehicles	Trust layer	Allocation stability	Real-time monitoring	Delivery success	Performance feedback
Torabbeigi et al. (2020) [23]	Linear regression	Payload	[23]	UAVs	×	×	×	×	×
Tseng et al. (2017) [32]	Linear regression	Motion, payload, and wind data	[32]	UAVs	×	×	×	×	×
Muli et al. (2022) [34]	LSTM	Device, task, and flight environment data	[35]	UAVs	×	×	×	×	×
Choudhry et al. (2021) [36]	Deep learning	Linear velocity, and payload	[35]	UAVs	×	×	×	×	×
El-Latif and El-dosuky (2025) [37]	Mathematical, and RF	Total weight, altitude, thrust, wind data	×	UAVs	×	×	×	×	×
Gao et al. (2024) [9]	Linear equation	Payload	×	Both	×	×	×	×	×
Abualola et al. (2023) [6]	Linear regression [23]	Payload	×	Both	×	✓	×	×	×
Zhang et al. (2023) [13]	×	×	×	GVs	×	✓	×	×	×
Triantali et al. (2025) [44]	×	×	×	GVs	×	✓	×	×	×
Elmay et al. (2025) [14]	×	×	×	GVs	✓	×	✓	✓	✓
Proposed model	RF	Device, task, flight, and environment data	[35]	Both	✓	✓	✓	✓	✓

decision support [17,26–30,53]. In cold-chain logistics, for example, digital twins monitor refrigerated goods by simulating thermal and mass-transfer phenomena via multiphysics models [53–55] or by employing data-driven machine learning models to predict spoilage under varying storage conditions [56].

In crowdsourced delivery, DTs facilitate decision-making by adapting fair task allocation and incentive mechanisms in real-time [57]. In [58], the authors integrate DTs into reinforcement-learning models to monitor and adapt participant strategies dynamically, while the work presented in [59] employ DTs for virtual representation of delivery courier behavior. In [14,47], the authors employ DTs to oversee the package status and storage conditions throughout the delivery duration and generate condition-based performance feedback, thereby enabling more accurate evaluation of worker reliability and delivery quality. Although DTs in supply chains and LMD are frequently discussed for planning and decision support in the literature, the majority of studies remain conceptual. In our work, we integrate a physics based DT model that accurately simulates packages behavior and status.

### 3. Proposed approach

This section presents our proposed blockchain-based LMD framework, which fuses predictive data analysis, DTs-driven behavior simulation, and a Gale–Shapley stable matching algorithm. By integrating these components in a secure, decentralized environment, the model aims to reduce delivery costs while improving both delivery time and quality.

#### 3.1. System overview

The framework consists of two main components: a set of delivery parcels modeled as tasks  $T$  and a set of hybrid crowdsourced workers

$W$ . Each task  $t \in T$  is identified by a tuple of  $\langle EA_j, r_j, PL_t, DL_t, r_t, D_t, TC_t \rangle$ , where  $EA_j$  and  $r_j$  denote the Ethereum Address (EA) and on-chain reputation of the requester;  $PL_t$  and  $DL_t$  specify pickup and drop-off coordinates;  $r_t$  is the reward value;  $D_t$  refers to the package's digital twin, which simulates behavior and condition over time; and  $TC_t$  is the time constraint of the task derived from the twin's delivery success profile. Similarly, every worker  $w_i \in W$  is identified by the tuple  $\langle EA_i, L_i, r_i, M_i \rangle$ , where  $EA_i$ ,  $L_i$ , and  $r_i$  represent the worker's EA, current location, and reputation, respectively, while  $M_i$  is a trained machine-learning model.

As illustrated in Fig. 1, the framework functions through three core phases — registration, allocation, and monitoring — executed through smart contracts and machine learning models to ensure secure, efficient, and reliable LMD.

1. **User Registration and Task Publishing:** UAV and GV workers register on-chain by submitting operational parameters, geolocation, and vehicle specifications. Requesters publish delivery tasks; each defined by pickup and drop-off coordinates, payload details, reward, and time constraints. All worker and task data are recorded on-chain via Ethereum smart contracts.
2. **ML-based Delivery Allocation:** Pre-trained ML models predict UAV energy requirements and GV's delivery success probabilities. These estimates generate Quality of Service (QoS) and Quality of Request (QoR) rankings for tasks and workers. A Gale–Shapley stable-matching algorithm then pairs tasks with workers. All model parameters and DT simulations are stored in IPFS for integrity and distributed access.
3. **Status Monitoring and Feedback:** In the post-allocation phase, each package's DT continuously captures environmental conditions, handling quality, and adherence to temporal constraints. Alerts are triggered upon deviations, and are used to compute delivery performance scores, directly influencing worker reputation updates.

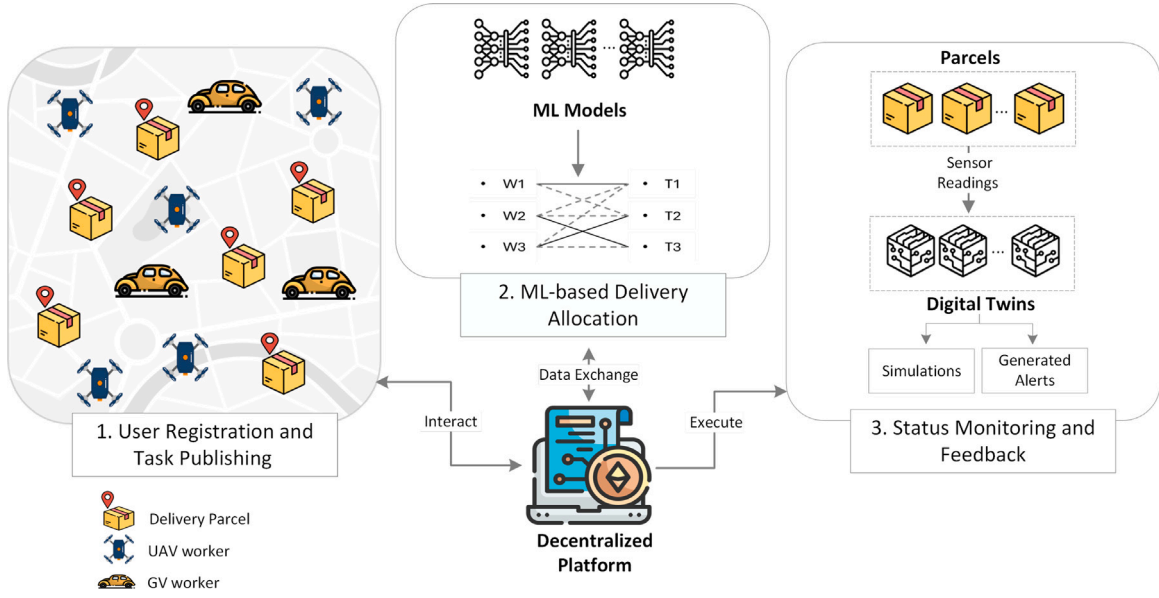


Fig. 1. Overview of the proposed machine learning and digital twins-based decentralized crowdsourcing system for LMD.

Table 2

Notations.

Symbol	Definition
$w_i$	Worker $i$
$r_i$	Reputation of $i$
$r_j$	Reputation of requester $j$
$t$	Task
$PL_t, DL_t$	Pickup and dropoff locations of $t$
$a_t$	Status of $t$
$TC_t$	Time-constraint of $t$
$r_t$	Reward value of $t$
$QoS_i(t)$	QoS of $i$ with respect to $t$
$QoR_i(t)$	QoR of $i$ with respect to $t$
$p_i^t$	Success probability of $i$ with respect to $t$
$\tau_i^t$	Total delivery time of $i$ for $t$
$QoD_i^t$	Quality of delivery of $i$ for completed task $t$
$ev_i$	Excursion violations generated for $t$
$ev_i$	Status violations generated for $t$
$travel\_time_i$	Total time required to complete $t$
$Energy\_Level_i$	Current energy level of $i$
$Energy\_R_i^t$	Required energy to complete $t$
$consumption_i^t$	Estimated consumption of $i$ with respect to $t$
$t\_completed_i$	Task $t$ completed by $i$
$t\_allocated_i$	Task $t$ allocated to $i$
$\gamma$	Weight factor
$proposal_i$	Proposals made by worker $i$
$matches$	Worker-task matches

The decentralized platform operates as the integration layer, enabling authenticated communication, executing smart contract logic, and maintaining a tamper-proof ledger of all platform activities throughout the delivery lifecycle. Table 2 provides the symbols used in this work. The components that make up the platform and their details are discussed in their respective sections below.

### 3.2. Framework users

#### 3.2.1. UAVs model: Proposed energy consumption prediction

The energy consumption model is designed to estimate the energy requirements of UAVs undertaking LMD tasks by accounting for dynamic flight conditions, vehicle orientation, payload mass, and environmental factors.

**Dataset Description** This study utilizes the experimental dataset generated in [35] collected using a *DJI Matrice 100* drone. The dataset

encompasses 209 test flights characterized by variations in altitude, speed, and payload. A total of 28 features were recorded during these test flights, including but not limited to battery state, device orientation, device location, and wind measurement. Of these, 21 features were used in training the consumption model, as detailed in Table 3. These features are categorized as follows:

- **Device-related features:** These pertain to the drone's operational parameters and include position, orientation, and motion metrics such as speed and acceleration.
- **Environment-related features:** Recognizing the influence of flight conditions on drone performance, this category includes features such as wind speed and direction.
- **Package-related features:** Represented by the mass of the carried payload.

**Machine Learning Model Architecture** Estimating UAV energy is framed as a regression problem, where the goal is to estimate the energy requirement for UAVs. The machine learning model selection process takes several critical factors into account. One key consideration is the model's ability to accommodate heterogeneous, noisy telemetry data and environmental inputs while remaining lightweight enough for on-chain execution. Smart-contract inference is both time- and gas-sensitive, so our model needs to deliver high accuracy without imposing heavy computational or cost overhead.

In light of these requirements, machine-learning models such as Random Forests (RF) and XGBoost are considered ideal due to their ability to manage heterogeneous data and reduce the risk of overfitting [60]. In this work, an RF regressor model is trained to predict energy consumption, as it takes into consideration the features' importance, scales efficiently, and achieves high accuracy. The RF model is built using the scikit-learn library, and specific hyperparameters were fine-tuned to further improve performance, as discussed in the following sections.

- $n\_estimators$  and  $max\_depth$ : the number of trees and the maximum depth of each tree, respectively.
- $max\_features$ : the number of features to consider when splitting.
- $min\_samples\_leaf$ : minimum number of samples at a leaf node.

Finally, the consumed energy of every UAV,  $consumption_{i=u}^t$  measured in  $\frac{kWh}{min}$ , is estimated as:

$$consumption_{i=u}^t = w_0 + x_1 w_1 + \dots + x_{21} w_{21}, \geq 0 \quad (1)$$

**Table 3**  
Dataset features used in training the energy consumption ML model.

Context	Feature	Description
Device	Time	Time elapsed in flight in seconds (s)
	Speed	Speed during a horizontal cruise in meters per second (m/s)
	Altitude	Altitude before following the delivery route in meters (m).
	Position X, Y, and Z	Longitude, latitude, and altitude in degrees deg.
	Velocity X, Y, and Z	Ground velocity components in meters per second (m/s).
	Angular X, Y, and Z	Angular velocity components in radians per second (rad/s).
	Orientation X, Y, Z, and W	Orientation in <i>quaternions</i>
	Linear acceleration X, Y, and Z	Ground acceleration in meters per second squared (m/s <sup>2</sup> )
Task	Payload	Weight of carried parcel in grams (g).
Environment	Wind speed	Airspeed in meters per second (m/s).
	Wind angle	Wind direction with respect to the north in degrees (deg).

where  $x = [x_1, x_2, \dots, x_{21}]$  is a feature vector from the considered dataset [35], including device-related features (position, velocity, orientation), environment-related features (wind speed and angle), and package-related features (payload weight). The weights  $w = [w_0, w_1, \dots, w_{21}]$  represent the learned weights of the RF model.

**Delivery Success Estimation** By incorporating the predicted consumption rate, the delivery success of a UAV worker for task  $t$ , denoted as  $p_{i=u}^t$ , is computed as shown in Eq. (2):

$$p_{i=u}^t = \begin{cases} 1 - \frac{\text{travel\_time}_u}{TC_t}, & \text{if } \text{Energy\_Level}_u \geq \text{Energy\_R}_u^t \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Here,  $\text{travel\_time}_{i=u}$  denotes the total flight duration required for the UAV to complete  $t$ , calculated based on its average speed and the Euclidean distance between pickup and drop-off points. The term  $TC_t$  represents the time constraint of the package as simulated by its DT, which will be detailed in Section 3.5.1. The delivery success probability  $p_{i=u}^t$  is defined as the complement of the ratio between  $\text{travel\_time}_{i=u}$  and  $TC_t$  given that the UAV has sufficient energy to complete  $t$ , as asserted using the condition  $\text{Energy\_Level}_u \geq \text{Energy\_R}_u^t$ . This ensures that the faster the UAV takes to deliver the package, the higher the probability of successful delivery.

As expressed in Eq. (3), the total energy requirement consists of the energy needed for both the pick up and delivery segments. The pickup flight assumes an empty payload and uses the consumption rate  $\text{consumption}_u^t(t_p = 0)$  along with the pickup flight duration  $\text{pickup\_time}_u^t$ . For the delivery segment, we use the consumption rate adjusted for the actual payload  $p$ ,  $\text{consumption}_u^t(t_p = p)$ , multiplied by the estimated delivery duration,  $\text{delivery\_time}_u^t$ .

$$\text{Energy\_R}_u^t = \text{consumption}_u^t(t_p = 0) \times \text{pickup\_time}_u^t + \text{consumption}_u^t(t_p = p) \times \text{delivery\_time}_u^t \geq 0 \quad (3)$$

### 3.2.2. GV's model: Delivery success prediction

The delivery success estimation models are based on the implementation of [14]. The estimation of safe delivery is structured as a classification problem. Due to the aforementioned requirements of a machine learning model in a decentralized platform, the architecture of these models is centered on RF classifiers. An RF classifier is trained for each worker on the platform based on their historical delivery data to enhance prediction accuracy.

The delivery success of every worker is estimated as follows:

$$p_{i=gv}^t = w_0 + z_1 w_1 + \dots + z_{28} w_{28}, \in [0, 1] \quad (4)$$

where  $z$  is a feature from the considered dataset [61].

### 3.3. Success prediction-based gale-shapley matching

The Gale-Shapely algorithm, introduced by Gale and Shapely, is a classical solution to the stable marriage problem and two-sided matching scenarios [62,63]. The algorithm guarantees stability, since no two participants would prefer to be matched with each other over

their current matches, and optimality as it produces the best possible outcome for the proposing side [64]. In our crowdsourced delivery context, stability ensures that no worker-task pair would mutually prefer each other over their assigned matches, preventing system disruption and ensuring long-term platform sustainability. Moreover, the algorithm's polynomial-time complexity  $O(n^2)$  makes it suitable for real-time deployment in blockchain environments.

This subsection formulates the task assignment as a matching game between two agent sets: workers  $W$  and tasks  $T$ . The allocation process is divided into  $n$  shifts, where each worker can undertake only one task per shift and each task is ultimately fulfilled by a single worker chosen from its eligible pool. Preferences on both sides are computed using their pre-trained ML models combined with QoS and QoR metrics, with the feature set tailored to worker type; GV's factor in behavioral, task-specific, and environmental variables, whereas UAVs are primarily governed by their energy levels.

Following the preferences computation, a one-to-one matching game is introduced, with further details provided in the subsequent sections.

#### 3.3.1. Preferences list computation

**Task Preferences** To generate each task's preference list, we evaluate every worker's suitability by computing a Quality of Service (QoS) score that integrates the task's specific constraints and individual behavior. Specifically, the QoS of a worker  $i$  to a task  $t$  is determined using the estimated probabilities described in Sections 3.2.1 and 3.2.2. With the predicted values  $p_i^t$ , the  $QoS_i^t$  is computed following Eq. (5).

$$QoS_i(i) = \frac{p_i^t \times r_i}{\tau_i^t}, \in [0, 1] \quad (5)$$

where  $r_i$  represents the worker's historical reputation, and  $\tau_i^t$  is the total travel time for worker  $i$  to go from their current location  $L_i$  to the pickup point  $PL_t$  and onward to the drop-off point  $DL_t$ , ensuring that workers with longer delivery times receive proportionally lower QoS scores.

Algorithm 1 further details the process of task preferences list computation. For a given task, the workers are filtered based on the estimated success probability,  $p_i^t$ . Once eligible workers are identified, their QoS is computed using Eq. (5) and appended to the task's preference list. Finally, the preferences list then sorted in descending order of  $QoS_i(i)$ , ensuring that workers most capable of completing their assignments efficiently receive priority in the matching process.

**Worker Preferences** Workers generate their own preference lists by computing a Quality of Request (QoR) score for each available task. This evaluation helps workers prioritize tasks that align with their goals and constraints.

The QoR of task  $t$  for worker  $i$  is evaluated by considering the requester's reputation  $r_j$ , the task's reward  $r_t$ , and the estimated delivery cost  $c_i^t$  as shown in Eq. (6):

$$QoR_i(t) = \frac{r_j \times r_t}{c_i^t}, \in [0, 1] \quad (6)$$

**Algorithm 1** Task Preference List Computation

---

```

1:  $PreferenceList_T \leftarrow \emptyset$ 
2: foreach task  $t$  in  $T$  do
3:    $preferences_t \leftarrow \emptyset$ 
4:   foreach worker  $i$  in  $W$  do
5:     if  $i$  is UAV then
6:        $consumption_i^t \leftarrow$  predicted value using Eq. (1)
7:       compute  $p_i^t$  using Eq. (2)
8:     else
9:       compute  $p_i^t$  using Eq. (4)
10:    if  $p_{i=u}^t \neq 0$  OR  $p_{i=gv}^t == 2$  then
11:      compute duration  $\tau_i^t$ 
12:      compute  $QoS_i(i)$  using Eq. (5)
13:       $preferences_{i,i} \leftarrow QoS_i(i)$ 
14:     $PreferenceList_T \leftarrow PreferenceList_T \cup preferences_i$ 
15: output  $PreferenceList_T$ 
16: end

```

---

The delivery cost  $c_i^t$  is calculated based on the energy price per unit  $energy\_price_i$  and estimated energy consumption  $Energy\_R_i^t$  with respect to a given task  $t$  as illustrated in Eq. (7).

$$c_i^t = energy\_price_i \times Energy\_R_i^t, \geq 0 \quad (7)$$

While the energy for UAVs  $Energy\_R_{i=u}^t$  is estimated using Eq. (3), the energy consumption for GVs  $Energy\_R_{i=gv}^t$  is calculated using Eq. (8).

$$Energy\_R_{i=gv}^t = \left( d_{L_i}^{PL_t} + d_{PL_t}^{DL_t} \right) \times consumption_{i=gv}, \geq 0 \quad (8)$$

where  $d_{L_i}^{PL_t}$  is the Euclidean distance between the worker's current location and the pickup point,  $d_{PL_t}^{DL_t}$  is the Euclidean distance between the start and end delivery points, and  $consumption_{i=gv}$  is GV's energy consumption rate, measured in  $\frac{L}{100 \text{ km}}$ .

**Algorithm 2** Worker Preference List Computation

---

```

1:  $PreferenceList_W \leftarrow \emptyset$ 
2: foreach worker  $i$  in  $W$  do
3:    $preferences_i \leftarrow \emptyset$ 
4:   foreach task  $t$  in  $T$  do
5:     if  $i$  is UAV AND  $t.payload \leq i.max\_weight$  then
6:        $consumption_i^t \leftarrow$  predicted value using Eq. (1)
7:     end
8:     compute  $c_i^t$  using Eq. (7)
9:     compute  $QoR_i(t)$  using Eq. (6)
10:     $preferences_{i,t} \leftarrow QoR_i(t)$ 
11:   $PreferenceList_W \leftarrow PreferenceList_W \cup preferences_i$ 
12: output  $PreferenceList_W$ 
13: end

```

---

Algorithm 2 further describes how each worker builds its own preferences list. First, tasks whose payload exceeds the worker's capacity, in the case of UAVs, are excluded due to UAV's maximum lift. For the tasks that satisfy this requirement, the QoR is computed using Eq. (6) and appended to the worker's preferences list. These QoR scores are then sorted in descending order to form the worker's preference list, ensuring that only feasible, high-value tasks are considered for assignment.

**3.3.2. Gale-shapley matching game**

Task assignments proceeds via the Gale-Shapley stable-matching algorithm presented in Algorithm 3, which reconciles the preference lists of tasks and workers to produce a two-sided stable outcome. Beginning

with all the unassigned workers, each worker proposes to their highest-ranked task. If a task is unclaimed and the proposing worker appears in its preference list, they form a temporary match. If the task is already matched, it compares the QoS scores of the current match and the new suitor, retaining the worker with highest QoS and releasing the other. Rejected workers then propose to their next preferred task, and the process repeats until every worker is either matched or has exhausted their preference list. This iterative matching guarantees that no worker-task pair would both prefer each other over their allocated match, thereby ensuring a stable and preference-respecting allocation.

**Algorithm 3** Proposed Allocation Game

---

```

1:  $matches_i \leftarrow \emptyset$ 
2:  $proposal_i \leftarrow \emptyset$ 
3:  $allocated\_pairs \leftarrow [t : \leftarrow \emptyset, t \in T]$ 
4: input  $W, T, PreferenceList_W, PreferenceList_T$ 
5: while  $i \in W$  is available AND  $preferences_i \neq \emptyset$  do
6:   foreach task  $t$  in  $preferences_i$  do
7:     if  $proposal_i = \emptyset$  then
8:        $proposal_i = t$ 
9:        $matches_i = allocated\_pairs[t]$ 
10:    if  $i \in preferences_t$  AND  $matches_t = \emptyset$  then
11:      if  $matches_t = \emptyset$  then
12:         $matches_t = i$ 
13:        break
14:      else
15:         $i_{prev} = matches_t$ 
16:        retrieve  $QoS_{i_{prev}}$  AND  $QoS_i$ 
17:        if  $QoS_{i_{prev}} < QoS_i$  then
18:           $matches_t = i$ 
19:         $allocated\_pairs[t] = matches_t$ 
20:      else
21:         $preferences_i = preferences_i - t$ 
22:    output  $allocated\_pairs$ 
23: end

```

---

**3.4. Decentralized platform**

The platform is implemented on the Ethereum blockchain and comprises two core smart contracts as described in Table 4: **UserContract** and **DeliveryContract**.

The primary role of the **UserContract** is to manage participants and their information through the following functions: **registerUser()** records worker and requester profiles, **addTask()** to publish delivery requests, **updateReputation()** to update the workers' reputations in the post-allocation step, as further explained in Section 3.5.2.

The **DeliveryContract** oversees the task allocation process using the following functions: **predict()** invokes ML models for predicting the GV success probability and UAV energy consumption; **taskPreferences()** and **workerPreferences()** compute the preference lists of both the tasks and workers, respectively, (Section 3.3.1); and **matching game()** executes the Gale-Shapley algorithm for stable one-to-one matching (Section 3.3.2). Lastly, **computeQoS()** leverages package DT data to compute the QoS metric used in performance feedback, which will be further explained in Section 3.5.1.

**3.5. Status monitoring and performance feedback****3.5.1. Package digital twin**

In the post-allocation phase, the framework monitors each package through its twin, which continuously mirrors the package's physical state and generates data-driven performance feedback. Unlike traditional tracking systems that primarily report geospatial data, the proposed twins simulate how environmental factors, such as temperature

**Table 4**  
Smart contract functions.

Functions	Input parameters
UserContract	
<i>registerUser()</i>	User information
<i>addTask()</i>	Task information
<i>updateReputation()</i>	QoD and delivery status
DeliveryContract	
<i>predict()</i>	T and W context
<i>computeQoD()</i>	Package DT output
<i>taskPreferences()</i>	W, T, inference values
<i>workerPreferences()</i>	W, T, inference values
<i>matching_game()</i>	Preference lists

changes, interact with a package's contents in real-time, allowing for early detection and mitigation of potential quality degradation. By translating these simulated conditions into delivery quality metrics, the system can assess worker performance on actual handling outcomes rather than mere task completion. Every twin update and simulation results are recorded immutably on blockchain, ensuring trusted transparency and traceability across all stakeholders.

To quantify the added value of digital twins, we introduce a vaccine-delivery use case in which each package twin employs the Arrhenius-based ODE model proposed in [65]. The model simulates the real-time degradation of vaccines, focusing on the relationship between storage temperature, initial state, and time, as detailed in Eq. (9).

$$S^t = \int_{t_{start}}^{t_{end}} \frac{\partial \alpha}{\partial t} (Temp^t, \alpha^t) \quad (9)$$

At each sensor interval; from  $t_{start}$  to  $t_{end}$ , the twin injects the current storage temperature  $Temp^t$  and impurity level  $\alpha^t$ , with the universal gas constant  $R = 8.314 \text{ J/(mol K)}$  to predict impurity progression. This continuous simulation triggers early violation alerts for the computation of objective QoD scores.

Additionally, the package's DT generates a delivery success profile (DSP) by simulating its impurity level over time as detailed in [14]. The DSP identifies the package's time constraint  $TC_i$ ; the elapsed duration until the impurity reaches a critical threshold, which is then used in building its preferences list as discussed in Section 3.2.1.

### 3.5.2. Workers performance feedback

The performance of the workers is evaluated by considering both delivery completion and quality. Given that the workers are assigned tasks from their preferred list, any cancellation or damage event results in a reputational penalty. Formally, after worker  $i$  completes or fails to complete the allocated task  $t$ , the new reputation  $r_{ij}$  is computed following Eq. (10):

$$r_{ij} = r_{ij-1} \times \frac{t_{completed_i}}{t_{allocated_i}} \times QoD_i^t, \in [0, 1] \quad (10)$$

where  $r_{ij-1}$  refers to worker  $i$ 's reputation immediately before the assignment,  $t_{completed_i}$  is a binary indicator of whether the worker completed or canceled the allocated task  $t_{completed_i}$ , and  $QoD_i^t$  represents the QoD score achieved by the worker on the completed task  $t$ .

$$QoD_i^t = e^{-(\gamma \times ev_i + [1-\gamma] \times sv_i)}, \in [0, 1] \quad (11)$$

The QoD of a package is computed by accumulating two alert types over the delivery window: excursion violations  $ev$  and status violations  $sv$ . Excursion violations occur when the storage conditions, such as temperature, fall outside the optimal range, while status violations reflect the package's response to these deviations, as predicted by the simulation. These aggregated violation values feed directly into Eq. (11), yielding a single metric that reflects storage compliance and predicted quality degradation.

## 4. Evaluation

This section presents a comprehensive analysis of our proposed framework through systematic sensitivity evaluation across multiple areas. We first detail the experimental setup, including simulation parameters, dataset characteristics, and ML models configuration. Then, we conduct a four-tiered analysis: (1) assessing the accuracy of our UAV energy consumption model against state-of-the-art benchmarks, (2) validating the real-time monitoring and behavior simulation capabilities of the digital-twin framework through comprehensive operational testing, (3) analyzing the performance of our allocation mechanism by varying the task load and UAV workers density, and (4) evaluating the feasibility and security implications of the blockchain component.

presents comprehensive operational validation of the digital twin framework, demonstrating real-time monitoring capabilities and mathematical model accuracy.

### 4.1. Setup

Table 5 summarizes the ML model training parameters and simulation environment used to evaluate our framework. The UAV energy model is trained on the dataset described in [35]. The dataset is split into 80% training and 20% testing sets. Key hyperparameters such as number of trees, maximum tree depth, and minimum samples per leaf are optimized via cross-validated randomized search to balance predictive accuracy.

For the broader simulation environment, we employ the *Mobility Uber Peru* dataset, which provides real-world trips with pickup and drop-off coordinates over a  $78 \times 93 \text{ km}^2$  grid [61]. The trips are modeled as delivery tasks. We build on the dataset to include at least 10% cancellations, uniformly distributed rewards, and payloads sampled between 50 g and 1 kg. The simulation consists a total of 400 tasks, used over 5 iterations, with each iteration containing 30–80 tasks.

The hybrid worker pool comprises 18 GVs and 12 UAVs. The initial reputation values for the workers are randomly assigned within a range of 0.3 to 0.6. Each GV worker is assigned a random speed between 40 to 80 km/h while their energy is varied between 1.7 to 5 L [6]. The delivery success model for GVs was trained following the parameters described in [14]. On the other hand, speed and energy levels of the UAVs are set according to the specifications of the considered drone, *DJI Matrice 100*. Flight speed varies from 36 km/h (the maximum speed while carrying the maximum possible payload) to 61 km/h (the maximum speed without any payload), and energy levels vary from 0 to 41 Wh. All the values are generated following a uniform distribution.

The evaluation experiments were conducted using Python 3.9 within Visual Studio Code version 1.88.1 on a PC with a 64-bit Windows 11 operating system, Intel Core i7 processor, and 16 GB of RAM. The smart contracts were implemented using Solidity 0.8.24 and compiled on Remix IDE version 0.62.1.

### 4.2. Proposed UAVs energy model evaluation

To validate our UAV energy-consumption predictor, we compare its performance against three representative benchmarks. The authors of [32] employ a linear optimization model to estimate energy consumption, while the authors of [34] leverage an LSTM network. In a more integrated setting, the authors of [6] utilize a statistical estimator for pre-assignment energy computation within a crowdsourced delivery platform. The comparison between the proposed model and existing works is detailed in Table 6.

To benchmark these models, we tested their energy-consumption predictions on two hold-out subsets from the dataset in [35]. Their performance was evaluated using the Mean Absolute Error (MAE) between predicted and observed consumption values. The MAE metric represents the average absolute errors and is, therefore, less sensitive

**Table 5**  
Simulation environment and ML models training setup.

UAV energy model training	
Dataset split	80% training, 20% testing
KFold, Epochs	10, 30
n_estimators	[50–200]
max_depth	[5–15]
max_features	15
min_samples_leaf	5
DTs attributes	
Initial impurity	0.2 mg/mL
Warning level	1 mg/mL
Critical level	1.5 mg/mL
Heat exposure	4 °C → 12 °C
Cold chain breaks	4 °C → −1 °C
Tasks attributes	
Available tasks	[30–80] tasks
Monitored condition	Impurity level mg/mL
Cancellation rate	[10–50] %
Reward budget	[5–20] \$
Weight	[0.05–1] kg
Workers attributes	
Workers	[4–18] GVs, [6–12] UAVs
Initial reputation	[0.3–0.6]
GVs speed	[40–80] km/h
UAVs speed	[36–61] km/h
GVs energy	[1.7–5] L
UAVs energy	[0–41] Wh
QoD weight factor $\gamma$	0.35

**Table 6**  
Comparison of existing energy models on flight 276's data of the same dataset.

Paper	Avg. MAE	UAV Model	Model
[32]	46.02	DJI Matrice 100	Non-linear regression
[6,23]	6.83	Phantom 4 Pro+	Mathematical
[34]	2.35	DJI Matrice 100	LSTM
Proposed	1.66	DJI Matrice 100	RF

to outliers [66]. As shown in the table, machine learning-based models outperform the computation-based estimators. Notably, the proposed model outperforms the rest with significantly low MAE values, achieving at least a 75% improvement relative to the statistical models of [6,32], and a 30% improvement over the LSTM-based model of [34]. This improvement arises from both the richer feature set compared to the computation-based models and the Random Forest's resilience to outliers and heterogeneity in contrast to the LSTM network.

#### 4.3. Proposed digital-twin evaluation

To validate the proposed DT model, we implemented the Arrhenius-based degradation model with realistic vaccine delivery scenarios. The experimental setup, as described in Table 2, utilized an initial impurity level of 0.2 mg/mL with a delivery time window of 180 min. The warning and critical levels were defined at 1.0 mg/mL and 1.5 mg/mL. The simulation environment also incorporated four temperature scenarios with stable storage, normal fluctuations, heat exposure events, and cold chain breaks. The values of the activation energy ( $E_a$ ), pre-exponential factor ( $A$ ), and gas constant ( $R$ ) were retrieved from the work discussed in [65]. The model a two-pathway Arrhenius degradation model, with activation energies of  $E_{a1}$  and  $E_{a2}$  set to 74.8 kJ/mol and 186.6 kJ/mol, pre-exponential factors  $A_1$  and  $A_2$  of  $e^{16.0} \text{ s}^{-1}$  and  $e^{59.4} \text{ s}^{-1}$ , and gas constant ( $R$ ) of 8.314 J/(mol × K).

Fig. 2 demonstrates the DT's real time violation detection capabilities the two-pathway Arrhenius degradation model through three representative delivery scenarios. The left panel shows temperature profiles for stable, heat exposure events, and cold chain breaks, while the right panel tracks corresponding QoD score evolution throughout

the 180 min delivery period. The stable storage scenario maintains temperature within the safe range of 2 °C–8 °C with minor fluctuations, resulting in a consistently high QoD score of 1. This represents optimal delivery conditions where both the excursion and status violations remain minimal.

The heat exposure event simulates events such as equipment failure, where the temperature rapidly increases from 4 °C to 12 °C and remains elevated for 20 min before recovery. The DT detects excursion immediately, causing the QoD score to decrease to 0.5 during the thermal event. Even after temperature recovery, the QoD score continues to decline due to the accumulated impurity damage, demonstrating the irreversible nature of thermal degradation.

The cold chain break scenario represents refrigeration failure or freezing conditions, with the temperature dropping to −1 °C for an extended duration. The QoD degradation follows a similar pattern to the heat exposure, with the score dropping from 1.0 to approximately 0.7. This similarity highlights how both extreme temperature deviations, whether hot or cold, trigger the two-pathway degradation mechanism. This real-time simulation capability enables delivery operators to implement immediate corrective actions during temperature excursions and allow a transparent and trusted monitoring for the requesters.

#### 4.4. Allocation mechanism evaluation

This section evaluates our allocation mechanism, combining the UAV energy predictions, delivery-success estimation, and DTs feedback.

##### 4.4.1. Benchmarks

The works discussed in [6,14] are selected as suitable benchmarks for evaluating the proposed mechanism. The work presented in [6] centers around stable task allocation using a centralized Gale-Shapley algorithm, which factors in the preferences of both tasks and hybrid workers; UAVs and GVs. Despite its preferences respecting assignments, the centralized design introduces security and trust vulnerabilities, lacks real-time package condition tracking after allocation, and evaluates workers only for delivery completion. On the other hand, the work proposed in [14] employs a greedy allocation mechanism for GV workers, whereby workers with highest QoS are selected. The work also integrates success prediction model trained on a public dataset, DTs for real-time package monitoring, and delivery quality-based worker evaluation. Although the system is implemented on a blockchain to ensure secure task allocation and data integrity, it does not consider the stability of the allocations and focuses entirely on GVs. For ease of reference, the benchmarks [6,14] are referred to as *GSM* and *Greedy*, respectively, throughout this section.

##### 4.4.2. Comparison metrics

The metrics used to compare the performance of the proposed system against the benchmarks are as follows:

- **QoS:** Average QoS of selected workers computed using Eq. (5). Higher values represent a better allocation mechanism.
- **QoR:** Average QoR of tasks that workers have selected, which is computed using Eq. (6). Higher values represent a better allocation mechanism.
- **Successfully Delivered Tasks %:** Percentage of allocated packages delivered with a QoD value of 1, which is computed using Eq. (11).
- **Reputation:** Average reputation of workers after the completion and/or cancellation of the allocated tasks. The individual values are computed using Eq. (10). Higher values indicate fewer cancellations and damaged deliveries.

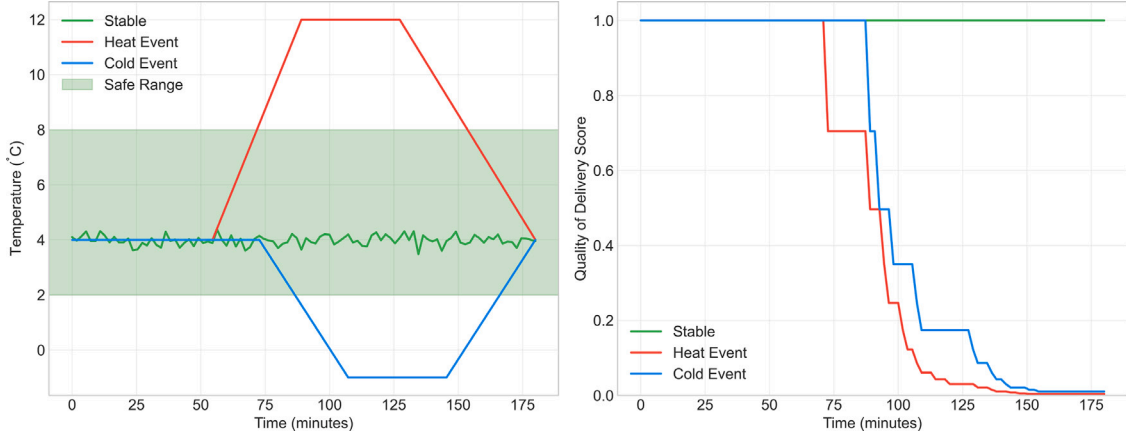


Fig. 2. Evaluation results of digital twin model with varying temperatures.

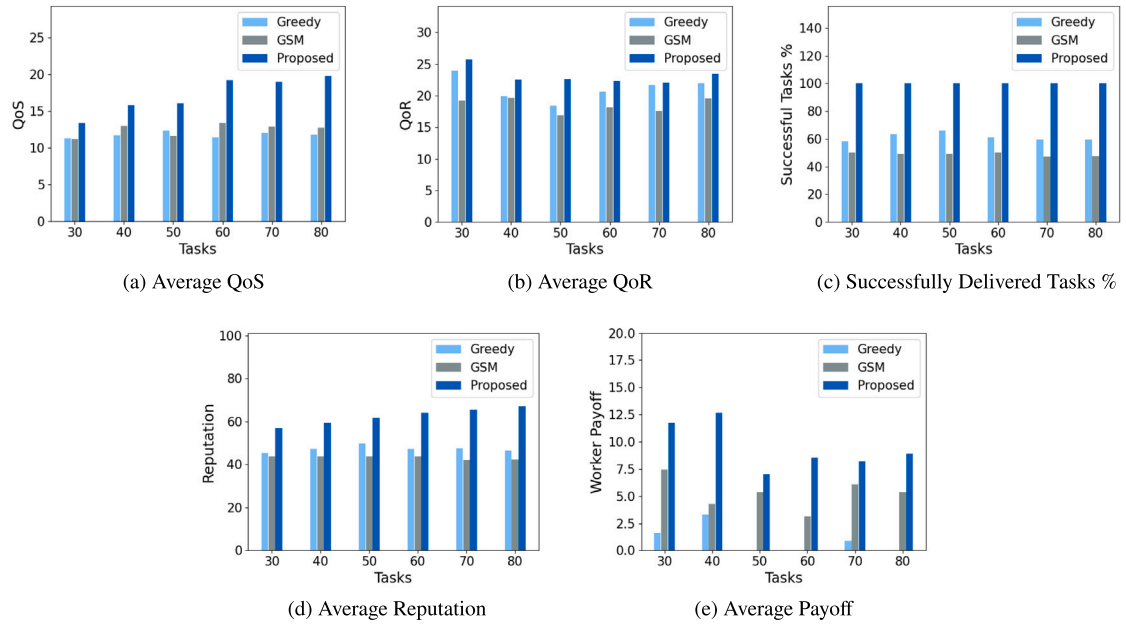


Fig. 3. Evaluation results of allocation mechanism with varying task load.

- **Worker Payoff:** Average reward of the selected workers after completing the delivery of the allocated tasks as shown in Eq. (12). Individual payoff values are computed using the defined reward of the given task  $re$  and the cost of delivery incurred by a worker, which is computed following Eq. (7).

$$Workers\_Payoff = \frac{\sum_i^n re_i - c_i^t}{n}, \geq 0 \quad (12)$$

Each data point is an average of 5 iterations, corresponding to a total runtime of five days for the platform.

#### 4.4.3. Varying task load evaluation

In this section, the platform is evaluated on its scalability by varying the task load between 30 and 80. The number of workers is fixed at 30, with 18 GVs and 12 UAVs. All the allocation mechanisms manage to allocate 100% of the tasks while their performances vary significantly across the defined metrics. The comparison results are shown in Fig. 3 and discussed in detail below.

Fig. 3(a) presents the average QoS of workers matched under each mechanism. Our proposed mechanism consistently outperforms both the GSM and Greedy benchmarks, showing an average improvement

of 37% and 46%, respectively. This significant improvement stems from our integrated approach to QoS computation, which incorporates delivery success probability predictions based on our precisely tuned energy consumption model. Additionally, Fig. 3(b) shows the QoR values of the tasks matched to workers. The proposed system outperforms both benchmarks, improving the QoR by an average of 10% compared to the Greedy and 25% compared to the GSM. This improvement is attributable to the integration of more accurate delivery cost estimation in the QoR computation, due to the improved energy consumption model, which accounts for environmental variables as well as multiple flight parameters, rather than the simplified payload-based energy model used by the GSM.

Fig. 3(c) shows the percentage of successfully delivered tasks; defined as completed deliveries with a perfect QoD score. The proposed mechanism consistently achieves a 100% in all the task loads, ensuring the successful delivery of all assigned tasks. This high rate is attributed to the integration of success predictions and realistic energy estimations in the matching process. On the other hand, the Greedy mechanism achieves an average of 60% despite integrating successful delivery predictions. The lower success rate is mainly due to unwanted task assignments, which often lead to cancellations. Similarly, the GSM achieves a 50% success rate due to the assignments of tasks to workers

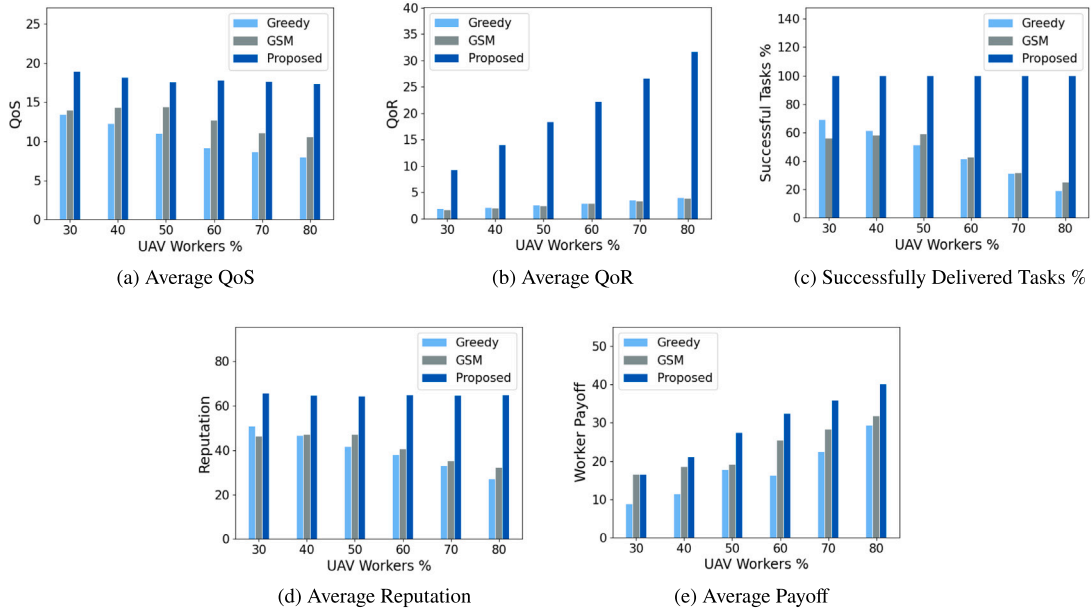


Fig. 4. Evaluation results of allocation mechanism with varying percentage of UAVs.

who are likely to damage them. overall, the proposed system improves the success rate by at least 53%.

Fig. 3(d) presents the average reputation of workers. As evident from the figure, the proposed mechanism drives reputation gains as task load grows, reflecting its 100% delivery success rate shown in Fig. 3(c). In contrast, the Greedy mechanism unfairly penalizes workers who cancel tasks despite not factoring in their preferences into the allocation process. In the case of GSM, even though it considers the workers' preferences and is able to fairly penalize worker cancellations, it is not able to prevent damaged deliveries. Hence, the proposed system improves workers' reputations by an average of 32% and 45% compared to the Greedy and GSM methods, respectively.

Finally, Fig. 3(e) shows the average worker payoff. The results indicate that the proposed mechanism achieves a significantly higher payoff compared to the benchmarks. The Greedy method, which does not consider the cost of delivery during task allocation, sees a decline or even zero payoff as the task load increases. On the other hand, while GSM yields lower payoff values compared to the proposed system. As shown in Table 6, GSM tends to underestimate UAV delivery costs as it relies on a simplistic energy model, which ultimately affects the total payoff for the workers. Overall, the proposed mechanism improves the total worker payoff by at least 38%, demonstrating its advantage in optimizing worker compensation through accurate cost and reward integration.

#### 4.4.4. UAV workers scalability evaluation

In this evaluation, we assess the impact of accurate cost estimations and UAV delivery efficiency by varying the percentage of UAV workers between 30% and 80%, within a fixed 20-worker pool. All the allocation mechanisms manage to allocate 100% of the tasks. The comparison results are illustrated in Fig. 4 and the findings are elaborated below.

Fig. 4(a) shows how the average QoS of selected workers evolves with increasing UAV density. All mechanisms exhibit a gradual decline in QoS values, an anticipated effect as UAVs have slower transit time compared to GVs. Despite this universal pattern, the proposed system shows a considerable enhancement in QoS, with an average improvement of 41% and 77% compared to the GSM and Greedy, respectively.

Fig. 4(b) presents the average QoR of the allocated tasks. As the number of UAVs increases, the QoR of the proposed system improves significantly. This is attributed to the increased selection of UAVs,

Table 7

Cost analysis of the smart contract functions when invoked on the public Ethereum network.

Function	Execution gas	ETH
<code>addTask()</code>	162 207	5.04E-4
<code>taskPreferences()</code>	139 320	4.33E-4
<code>workerPreferences()</code>	126 996	3.94E-4
<code>matching_game()</code>	1 034 293	3.21E-3

which reduces delivery costs, and, consequently enhances QoR according to Eq. (6). On the other hand, the benchmarks show minimal change in performance across varying UAV ratios, indicating lower selection of UAVs compared to GVs.

Fig. 4(c) illustrates the ratio of successfully completed tasks to the allocated tasks. The proposed system consistently achieves a 100% success rate, whereas both benchmarks exhibit marked declines: GSM's success falls by at least 45% and Greedy's by 79%, due to their simpler energy estimators underestimate UAV consumption, leading to unfinished or damaged shipments. These shortfalls directly impact worker reputations, as shown in Fig. 4(d), where both benchmarks see reputation drops with increasing number of UAVs. Conversely, the proposed system maintains higher reputation values, with an average improvement of 60% and 71% in comparison to the GSM and Greedy, respectively.

Finally, the worker payoff is shown in Fig. 4(e). Although all mechanisms exhibit increased payoffs as the UAV numbers grow, the proposed system outperforms the benchmarks by an average of 22% compared to the GSM and 70% compared to the Greedy.

#### 4.5. Blockchain evaluation

To assess on-chain feasibility, we measured the gas consumption of all core smart-contract functions by deploying them on Remix IDE. The cost is primarily determined by the gas consumed, which depends on the complexity and size of the functions, as well as the variables and data structures used. Table 7 summarizes the cost breakdown for invoking the functions, including the execution gas and its fiat equivalent in ethers, Ethereum blockchain's native token. Each function's gas usage was recorded and converted into ETH cost using a gas price of 3.10 Gwei as of April 23, 2025 [67].

As shown in Table 7, creating a task on the platform incurs a cost of  $5.04\text{E}-4$  ethers for the requester. Submitting a preference for a task costs approximately  $4.33\text{E}-4$  ethers per worker while the cost of preference submission by a worker is estimated at  $3.94\text{E}-4$  ethers per task. Finally, executing the matching game for one task with a preference list of 10 workers incurs a cost of  $3.21\text{E}-3$  ethers.

The integration of blockchain technology enhances the platform's transparency and security. Smart contracts automate task allocations and data updates while preserving authenticity, as each action is cryptographically recorded on-chain. In particular, our blockchain layer delivers:

- **Availability:** Package DT data and delivery records are stored across a distributed network, eliminating single points of failure or reliance on third-party servers. Authorized participants can retrieve historical and real-time information without risking data loss or unauthorized alteration.
- **Immutability:** Every smart-contract invocation, whether registering a user, publishing a task, or updating reputation, generates a transaction log. Once committed, these entries cannot be altered or deleted, guaranteeing that all delivery events remain permanently verifiable.
- **Accountability and Non-repudiation:** Every participant operates under a unique Ethereum address and every transaction is signed by the private key of the invoker and broadcast to the network. Hence, a user cannot deny a given action, which is prevalent in the delivery of sensitive packages with crowdsourced workers.

#### 4.6. Practicality and limitations

The performance improvements demonstrated by our framework carry substantial practical implications for both LMD systems and broader logistics operations. First, the observed gains in quality of service (QoS) and successful delivery rates with the integration of UAVs directly address core industry concerns, namely reliability and efficiency. Second, DTs further transform quality assurance and traceability for temperature- and storage-sensitive goods. For instance, pharmaceutical providers could leverage our framework to maintain continuous cold-chain monitoring with verifiable records; a regulatory requirement that current systems struggle to provide cost-effectively. The real-time condition monitoring enables proactive interventions before product degradation occurs, potentially saving millions in spoilage costs annually. Similar applications could also be discovered in other perishables such as agricultural produce deliveries. Third, by enhancing the average worker payoff without compromising service standards, the proposed approach considerably strengthens the value proposition of crowdsourced delivery platforms.

Despite its demonstrated advantages, our framework faces several technical limitations. The growing adoption of blockchain technology by both individuals and organizations has raised network scalability concerns. One of the notable approaches in mitigating scalability issues is Ethereum's transition from a Proof of Work (PoW) to a Proof of Stake (PoS) consensus mechanism. This shift is aimed at improving the network's scalability and reducing congestion by facilitating greater transaction throughput. Additionally, layer 2 (L2) solutions like zkSync present a viable method for addressing scalability challenges [68]. L2 solutions process transactions off the main Ethereum network (Layer 1), rolling multiple transactions into a single batch [69], which increases scalability while lowering costs and enhancing overall efficiency.

Smart contract immutability, while beneficial for data integrity, presents a unique challenge. Once deployed, smart contracts cannot be altered, making careful logic design and optimization crucial. Moreover, since smart contracts frequently handle financial assets, they face an unprecedented number of attacks and exploits by malicious actors [70]. To safeguard against such threats, various tools help analyze the smart contract code to identify potential security flaws, ensuring these contracts are secure and perform as intended.

## 5. Conclusion

In this article, we introduce a blockchain-based crowdsourcing platform that combines the capabilities of UAVs and GVs for last-mile delivery. The proposed system maximizes the successful delivery of packages by leveraging a stable allocation mechanism, digital twins and machine learning models. The allocation mechanism is based on Gale-Shapely matching game that considers the preferences of both the tasks and workers for stable assignments. The platform uses key metrics such as QoS and QoR to compute preferences, integrating machine learning models that predict the consumption energy of UAVs and the likelihood of successful deliveries. The framework outperforms existing benchmarks across various metrics; QoS by at least 37%, QoR by at least 10%, and successfully completed tasks by at least 45%. The integration of worker preferences also supports fair penalization of workers who cancel tasks, eliminating unjust reputation decreases. As a result, the average worker reputation and payoff improve by at least 32% and 22%, respectively, compared to the existing systems. These improvements offer significant practical benefits for last-mile delivery operations by directly addressing key challenges in the industry — namely reliability, quality assurance, cost-effectiveness, transparency and accountability. Looking ahead, the framework can be expanded to address the end-to-end UAVs delivery process, the adaptation of various dynamic digital twins, and scalable blockchain implementation by exploring layer 2 solutions.

#### CRedit authorship contribution statement

**Feruz Elmay:** Writing – original draft, Software, Methodology, Conceptualization. **Maha Kadadha:** Writing – review & editing, Supervision, Conceptualization. **Shakti Singh:** Writing – review & editing, Supervision, Conceptualization. **Rabeb Mizouni:** Writing – review & editing, Supervision, Conceptualization. **Hadi Otrouk:** Writing – review & editing, Supervision, Conceptualization. **Azzam Mourad:** Writing – review & editing, Supervision, Conceptualization.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgment

This work was supported by the Center for Cyber-Physical Systems, Khalifa University, United Arab Emirates.

#### Data availability

The data used is publicly available.

## References

- [1] M. Barbosa, J.P. Pedroso, A. Viana, A data-driven compensation scheme for last-mile delivery with crowdsourcing, *Comput. Oper. Res.* 150 (2023) 106059.
- [2] M. Placek, Global last mile delivery market size 2020–2027 | Statista, 2023, URL <https://www.statista.com/statistics/1286612/last-mile-delivery-market-size-worldwide>, [Online; (Accessed 30 September 2024)].
- [3] World Economic Forum, The future of the last-mile ecosystem, 2020, URL [https://www3.weforum.org/docs/WEF\\_Future\\_of\\_the\\_last\\_mile\\_ecosystem.pdf](https://www3.weforum.org/docs/WEF_Future_of_the_last_mile_ecosystem.pdf), [Online; (Accessed 30 September 2024)].
- [4] D. Coppola, Topic: E-commerce logistics, 2024, URL <https://www.statista.com/topics/11962/e-commerce-logistics/#topicOverview>, [Online; (Accessed 30 September 2024)].
- [5] J. Kim, S.J. Kweon, S.W. Hwang, S. Lee, Crowdsourcing integration on the last mile delivery platform considering floating population data, *Expert Syst. Appl.* 248 (2024) 123312.

- [6] H. Abualola, R. Mizouni, H. Otrok, S. Singh, H. Barada, A matching game-based crowdsourcing framework for last-mile delivery: Ground-vehicles and unmanned-aerial vehicles, *J. Netw. Comput. Appl.* 213 (2023) 103601.
- [7] N. Elsokkary, H. Otrok, S. Singh, R. Mizouni, H. Barada, M. Omar, Crowdsourced last mile delivery: Collaborative workforce assignment, *Internet Things* 22 (2023) 100692.
- [8] A. Ermagun, A. Punel, A. Stathopoulos, Shipment status prediction in online crowd-sourced shipping platforms, *Sustain. Cities Soc.* 53 (2020) 101950.
- [9] J. Gao, Q. Wang, X. Zhang, J. Shi, X. Zhao, Q. Han, Y. Pan, Cooperative air-ground instant delivery by UAVs and crowdsourced taxis, in: 2024 IEEE 40th International Conference on Data Engineering, ICDE, IEEE, 2024, pp. 4153–4166.
- [10] Y. Zhao, K. Zheng, H. Yin, G. Liu, J. Fang, X. Zhou, Preference-aware task assignment in spatial crowdsourcing: from individuals to groups, *IEEE Trans. Knowl. Data Eng.* 34 (7) (2020) 3461–3477.
- [11] M. Safran, D. Che, Real-time recommendation algorithms for crowdsourcing systems, *Appl. Comput. Inform.* 13 (1) (2017) 47–56.
- [12] S. Yang, K. Han, Z. Zheng, S. Tang, F. Wu, Towards personalized task matching in mobile crowdsensing via fine-grained user profiling, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 2411–2419.
- [13] N. Zhang, Z. Liu, F. Li, Z. Xu, Z. Chen, Stable matching for crowdsourcing last-mile delivery, *IEEE Trans. Intell. Transp. Syst.* 24 (8) (2023) 8174–8187.
- [14] F. Elmay, M. Kadadha, R. Mizouni, S. Singh, A. Mourad, H. Otrok, Predictive safe delivery with Machine Learning and digital twins collaboration for decentralized crowdsourced systems, *J. Netw. Comput. Appl.* (2025) 104196.
- [15] M. Abououf, S. Singh, H. Otrok, R. Mizouni, E. Damiani, Machine Learning in mobile crowd sourcing: A behavior-based recruitment model, *ACM Trans. Internet Technol. (TOIT)* 22 (1) (2021) 1–28.
- [16] M. Kadadha, R. Mizouni, S. Singh, H. Otrok, A. Mourad, Crowdsourced vehicles and UAVs for last-mile delivery application using blockchain-hosted matching mechanism, *Veh. Commun.* (2024) 100761.
- [17] A. Sharma, S.K. Singh, S. Kumar, R. Thakur, B.B. Gupta, V. Arya, IoT-enabled smart farming with Industry 5.0, *J. High Speed Netw.* 30 (3) (2024) 477–496.
- [18] J. Daleo, WingNewPlane-Image1, *FLYING Mag.* (2024) URL <https://www.flyingmag.com/googles-wing-introduces-new-delivery-drone-with-double-the-payload>.
- [19] H. Chen, Z. Hu, S. Solak, Improved delivery policies for future drone-based delivery systems, *European J. Oper. Res.* 294 (3) (2021) 1181–1201.
- [20] S.R.R. Singireddy, T.U. Daim, Technology roadmap: Drone delivery–amazon prime air, *Infrastruct. Technol. Manag.: Contrib. from Energy Heal. Transp. Sect.* (2018) 387–412.
- [21] Y. Pan, S. Li, Q. Chen, N. Zhang, T. Cheng, Z. Li, B. Guo, Q. Han, T. Zhu, Efficient schedule of energy-constrained UAV using crowdsourced buses in last-mile parcel delivery, *Proc. the ACM Interact. Mob. Wearable Ubiquitous Technol.* 5 (1) (2021) 1–23.
- [22] S. Choudhury, K. Solovey, M.J. Kochenderfer, M. Pavone, Efficient large-scale multi-drone delivery using transit networks, *J. Artificial Intelligence Res.* 70 (2021) 757–788.
- [23] M. Torabbeigi, G.J. Lim, S.J. Kim, Drone delivery scheduling optimization considering payload-induced battery consumption rates, *J. Intell. Robot. Syst.* 97 (2020) 471–487.
- [24] X. Deng, M. Guan, Y. Ma, X. Yang, T. Xiang, Vehicle-assisted uav delivery scheme considering energy consumption for instant delivery, *Sensors* 22 (5) (2022) 2045.
- [25] M. Grieves, J. Vickers, Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems, *Transdiscipl. Perspect. Complex Syst.: New Find. Approaches* (2017) 85–113.
- [26] Y. Liu, S. Pan, P. Folz, F. Ramparany, S. Bolle, E. Ballot, T. Coupaye, Cognitive digital twins for freight parking management in last mile delivery under smart cities paradigm, *Comput. Ind.* 153 (2023) 104022.
- [27] A. Belfadel, S. Hörl, R.J. Tapia, J. Puchinger, Towards a digital twin framework for adaptive last mile city logistics, in: 2021 6th International Conference on Smart and Sustainable Technologies, SpliTech, IEEE, 2021, pp. 1–6.
- [28] H. Haße, B. Li, N. Weissenberg, J. Cirullies, B. Otto, Digital twin for real-time data processing in logistics, in: *Artificial Intelligence and Digital Transformation in Supply Chain Management: Innovative Approaches for Supply Chains. Proceedings of the Hamburg International Conference of Logistics, HICL, Vol. 27*, Berlin: epubli GmbH, 2019, pp. 4–28.
- [29] S.R. Jeremiah, L.T. Yang, J.H. Park, Digital twin-assisted resource allocation framework based on edge collaboration for vehicular edge computing, *Future Gener. Comput. Syst.* 150 (2024) 243–254.
- [30] S. Qanazi, E. Leclerc, P. Bosredon, Integrating social dimensions into urban digital twins: A review and proposed framework for social digital twins, *Smart Cities* 8 (1) (2025) 23.
- [31] L. Di Puglia Pugliese, F. Guerriero, M.G. Scutellà, The last-mile delivery process with trucks and drones under uncertain energy consumption, *J. Optim. Theory Appl.* 191 (1) (2021) 31–67.
- [32] C.-M. Tseng, C.-K. Chau, K.M. Elbassioni, M. Khonji, Flight tour planning with recharging optimization for battery-operated autonomous drones, 2017, *CoRR*, abs/1703.10049.
- [33] J. Zhang, J.F. Campbell, D.C. Sweeney II, A.C. Hupman, Energy consumption models for delivery drones: A comparison and assessment, *Transp. Res. Part D: Transp. Environ.* 90 (2021) 102668.
- [34] C. Muli, S. Park, M. Liu, A comparative study on energy consumption models for drones, in: *Global IoT Summit*, Springer, 2022, pp. 199–210.
- [35] T.A. Rodrigues, J. Patrikar, A. Choudhry, J. Feldgoise, V. Arcot, A. Gahlaut, S. Lau, B. Moon, B. Wagner, H.S. Matthews, et al., In-flight positional and energy use data set of a DJI Matrice 100 quadcopter for small package delivery, *Sci. Data* 8 (1) (2021) 155.
- [36] A. Choudhry, B. Moon, J. Patrikar, C. Samaras, S. Scherer, Cvar-based flight energy risk assessment for multirotor uavs using a deep energy model, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 262–268.
- [37] E.I.A. El-Latif, M. El-dosuky, Predicting power consumption of drones using explainable optimized mathematical and Machine Learning models, *J. Supercomput.* 81 (5) (2025) 646.
- [38] R. D'Andrea, Guest editorial can drones deliver? *IEEE Trans. Autom. Sci. Eng.* 11 (3) (2014) 647–648.
- [39] R. Bruns, J. Dötterl, J. Dunkel, S. Ossowski, A collaborative approach to mobile crowdsourcing based on data stream learning, in: *International Conference on Practical Applications of Agents and Multi-Agent Systems*, Springer, 2022, pp. 83–94.
- [40] T. Rashid, S. Anwar, M.A. Jaffar, H. Hakami, R. Baashirah, Q. Umer, Success prediction of crowdsourced projects for competitive crowdsourced software development, *Appl. Sci.* 14 (2) (2024) 489.
- [41] J. Zhang, X. Zhang, Multi-task allocation in mobile crowd sensing with mobility prediction, *IEEE Trans. Mob. Comput.* 22 (2) (2021) 1081–1094.
- [42] Y. Yang, W. Liu, E. Wang, J. Wu, A prediction-based user selection framework for heterogeneous mobile crowdsensing, *IEEE Trans. Mob. Comput.* 18 (11) (2018) 2460–2473.
- [43] M. Kadadha, H. Otrok, R. Mizouni, S. Singh, A. Ouali, On-chain behavior prediction Machine Learning model for blockchain-based crowdsourcing, *Future Gener. Comput. Syst.* 136 (2022) 170–181.
- [44] D.G. Triantali, K. Skouri, K.E. Parsopoulos, Probabilistic crowdshipping model for last-mile delivery, *Int. J. Syst. Sci.: Oper. Logist.* 12 (1) (2025) 2450051.
- [45] M. Hribernik, K. Zero, S. Kummer, D.M. Herold, City logistics: Towards a blockchain decision framework for collaborative parcel deliveries in micro-hubs, *Transp. Res. Interdiscip. Perspect.* 8 (2020) 100274.
- [46] S. El Moudaa, Y. Ibrahim, M. Kadadha, R. Mizouni, H. Otrok, S. Singh, Packchain: Toward a blockchain-based management platform for last-mile delivery, in: 2022 International Wireless Communications and Mobile Computing, IWCMC, IEEE, 2022, pp. 919–924.
- [47] F. Elmay, M. Kadadha, R. Mizouni, S. Singh, H. Otrok, A. Mourad, Digital twins and dynamic NFTs for blockchain-based crowdsourced last-mile delivery, *Inf. Process. Manage.* 61 (4) (2024) 103756.
- [48] L. Gao, T. Cheng, L. Gao, [Tswcrowd]: A decentralized task-select-worker framework on blockchain for spatial crowdsourcing, *IEEE Access* 8 (2020) 220682–220691.
- [49] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008, URL <https://bitcoin.org/bitcoin.pdf>, [Online; (Accessed 9 June 2025)].
- [50] N. Chaudhry, M.M. Yousaf, Consensus algorithms in blockchain: Comparative analysis, challenges and opportunities, in: 2018 12th International Conference on Open Source Systems and Technologies, ICOSST, IEEE, 2018, pp. 54–63.
- [51] V. Buterin, Ethereum white paper: A next generation smart contract & decentralized application platform, 2013, URL <https://github.com/ethereum/wiki/wiki/White-Paper>.
- [52] A.S. Shuaibu, A.S. Mahmoud, T.R. Sheltami, A review of last-mile delivery optimization: Strategies, technologies, drone integration, and future trends, *Drones* 9 (3) (2025) 158.
- [53] T. Defraeye, G. Tagliavini, W. Wu, K. Prawiranto, S. Schudel, M.A. Kerisima, P. Verboven, A. Bühlmann, Digital twins probe into food cooling and biochemical quality changes for reducing losses in refrigerated supply chains, *Resour. Conserv. Recycl.* 149 (2019) 778–794.
- [54] L. You, S. Schudel, T. Defraeye, Developing of biophysical food for monitoring postharvest supply chains for avocado and potato and deploying of biophysical apple, *J. Food Eng.* 338 (2023) 111219.
- [55] G. Tagliavini, T. Defraeye, J. Carmeliet, Multiphysics modeling of convective cooling of non-spherical, multi-material fruit to unveil its quality evolution throughout the cold chain, *Food Bioprod. Process.* 117 (2019) 310–320.
- [56] E. Bottani, G. Vignali, G.P.C. Tancredi, A digital twin model of a pasteurization system for food beverages: Tools and architecture, in: 2020 IEEE International Conference on Engineering, Technology and Innovation, ICE/ITMC, IEEE, 2020, pp. 1–8.
- [57] H.T. Ignatius, R. Bahsoon, Equity, equality, and need: Digital twin approach for fairness-aware task assignment of heterogeneous crowdsourced logistics, *IEEE Trans. Comput. Soc. Syst.* (2023).
- [58] L. Zhang, X. Wang, H. Lin, M.J. Piran, A crowdsourcing logistics solution based on digital twin and four-party evolutionary game, *Eng. Appl. Artif. Intell.* 130 (2024) 107797.

- [59] F. Yu, G. Zhang, H. Wang, D. Jin, Y. Li, Fine-grained courier delivery behavior recovery with a digital twin based iterative calibration framework, *ACM Trans. Intell. Syst. Technol.* 15 (5) (2024) 1–25.
- [60] A. Singh, N. Thakur, A. Sharma, A review of supervised Machine Learning algorithms, in: 2016 3rd International Conference on Computing for Sustainable Global Development, *INDIACom*, IEEE, 2016, pp. 1310–1315.
- [61] M. Russo, Mobility Uber Perú dataset, 2019, URL <https://www.kaggle.com/datasets/marcusrb/uber-peru-dataset>, [Online; (Accessed 24 April 2024)].
- [62] D. Gale, L.S. Shapley, College admissions and the stability of marriage, *Am. Math. Mon.* 120 (5) (2013) 386–391.
- [63] A.E. Roth, Deferred acceptance algorithms: History, theory, practice, and open questions, *Int. J. Game Theory* 36 (2008) 537–569.
- [64] I. Saglam, The success of the deferred acceptance algorithm under heterogenous preferences with endogenous aspirations, *Comput. Econ.* 57 (2) (2021) 577–591.
- [65] D. Clénet, F. Imbert, P. Probeck, N. Rahman, S.F. Ausar, Advanced kinetic analysis as a tool for formulation development and prediction of vaccine stability, *J. Pharm. Sci.* 103 (10) (2014) 3055–3064.
- [66] D.S.K. Karunasingha, Root mean square error or mean absolute error? Use their ratio as well, *Inform. Sci.* 585 (2022) 609–629.
- [67] Etherscan.io, Ethereum Gas Tracker | Etherscan, 2025, URL <https://etherscan.io/gasTracker>, [Online; (Accessed 21 January 2025)].
- [68] R. Neiheiser, G. Inácio, L. Rech, C. Montez, M. Matos, L. Rodrigues, Practical limitations of ethereum's layer-2, *IEEE Access* 11 (2023) 8651–8662.
- [69] A. Gangwal, H.R. Gangavalli, A. Thirupathi, A survey of layer-two blockchain protocols, *J. Netw. Comput. Appl.* 209 (2023) 103539.
- [70] S.S. Kushwaha, S. Joshi, D. Singh, M. Kaur, H.-N. Lee, Ethereum smart contract analysis tools: A systematic review, *IEEE Access* 10 (2022) 57037–57062.