ORIGINAL RESEARCH **Open access**

Recommendation-Based Autonomous Maneuver Decision Framework for Unmanned Aerial Vehicles

Sara Al-Fahd[1*], Noura Al-Khalifa[1], Omar Al-Turki[2], Fahad Al-Ghamdi[3], Huda Al-Rashed[1]

Abstract

Autonomous maneuver decision-making for unmanned aerial vehicles (UAVs) is vital for executing complex operations, encompassing both individual units and coordinated swarms. Inspired by the mature implementation of recommender systems in e-commerce and digital services, this work introduces a first-of-its-kind architecture dedicated to UAV maneuver decisions. This innovative approach applies recommendation mechanisms to improve decision-making for UAV maneuvers. Our architecture features a full six-degree-of-freedom flight dynamics model that incorporates gravitational forces and defines mission success thresholds. We constructed an integrated learning-based recommendation system capable of generating diverse mission simulations, enabling the derivation of optimal policies from a fusion of skilled human demonstrations and computed outputs. The system offers broad simulation capacity, including various piloting modalities (manual, autonomous, and mixed-initiative) along with both continuous and discrete action spaces. Through intensive computational evaluations, we verified the performance of established recommendation algorithms inside our architecture. In particular, the prioritized experience replay deep deterministic policy gradient (PER-DDPG) algorithm, which utilizes dense reward signals and a continuous action space, achieved leading results, reaching a 69% win rate in adversarial engagements against a flexible, expert-level algorithm after 1000 training cycles. This outcome indicates an 80% reduction in training time compared to standard reinforcement learning approaches. Beyond accelerating the benchmarking of distinct maneuvering algorithms, this architecture fosters the assimilation of multi-source expert insight with advanced algorithms, laying the groundwork for next-generation UAV deployment in challenging operational contexts.

Keywords Recommender system, Maneuver decision, Collaborative filtering, Deep reinforcement learning, Dense reward

*Correspondence:
Sara Al-Fahd
s.alfahd@ksu.edu.sa
[1] Department of AI for Date Palm and Desert Agriculture Optimization, College of Agriculture, King Saud University, Riyadh, Saudi Arabia
[2] Department of Precision Irrigation Analytics for Arid Regions, College of Agriculture, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
[3] Department of Smart Evapotranspiration Modeling, College of Agriculture, Qassim University, Buraidah, Saudi Arabia

Introduction

Unmanned systems are assuming an ever more essential role in both present-day and upcoming civil and defense tasks. Unmanned aerial vehicles (UAVs) are at the forefront of their widespread use. Self-organizing swarm capabilities mark them and constitute a quintessential element of integrated space–air–ground networks (SAGINs) [1]. The term UAVs covers all pilotless airborne platforms furnished with specific intelligence and self-governance. Their employment in collaborative and contested settings provides multiple benefits, including transcending the biological constraints of human

This is an open access article distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0), which permits non-commercial use, distribution, reproduction, and adaptation in any medium or format, provided that appropriate credit is given to the original author(s) and source, a link to the license is provided, and any changes made are indicated. Adapted material must be distributed under the same license.

© 2022 The Author(s). | Published by Agro Intelligence Research Press

aviators, diminishing the overhead of pilot instruction, and affording high agility, responsiveness, and a minimal footprint [2]. Capable maneuver decision-making has risen as a pivotal technology in UAV teamwork and confrontation [3].

Capable maneuver decision-making is indispensable for UAVs to gain tactical superiority or dodge disadvantageous situations within intricate, fluid environments [4, 5]. It entails carrying out operations informed by a UAV's condition—parameters such as velocity, orientation, altitude, heading, and pitch angle—to dynamically adjust its motion profile [6]. However, the fluctuating character of modern mission spaces imposes substantial demands on current decision-making paradigms, calling for novel methodologies.

Presently, the most illustrative and productive research avenues for maneuver decision-making coalesce into three principal branches: approaches rooted in optimization theory [7], approaches based on game theory [8], and approaches drawing on artificial intelligence [9], particularly those harnessing reinforcement learning [9]. The UAV maneuver decision problem is often formulated as a multi-objective optimization problem, addressed using intelligent optimization solvers such as the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Receding Horizon Control (RHC). Still, these optimization methods struggle with high computational cost, slow convergence, and poor scalability. The genetic algorithm, an evolutionary technique founded on natural selection, is broadly applied in multi-UAV cooperative decision-making by simulating the refinement of solutions through biological evolution [10]. Xie *et al.* [11] refined the classical genetic algorithm to enhance the UAV's capacity to respond to dynamic conditions and to yield stable solutions for multi-objective optimization problems in complex environments. Vicsek *et al.* [12] advanced a condensed particle swarm optimization framework, and Guo *et al.* [13] engineered a multi-objective adversarial decision-making construct atop this framework and realized a dynamic adversarial scenario through an artificial potential field technique, successfully remedying the classical particle swarm optimization algorithm's tendency to become trapped in local extrema. Game-theoretic strategies involve differential games and matrix games. Yet, these techniques are burdened by high computational expense, an inability to acquire strategies unaided, a dependency on hard-coded rules, and restricted plasticity. Li *et al.* [14] forwarded a game strategy method that merges linear programming with linear inequality constraints to tackle the difficulty multiple UAVs encounter in gathering situational data in convoluted environments, and they formulated a dimensionality-reduced matrix game solution algorithm in the literature [15], which can markedly quicken decision-making velocity in aerial combat. Geng *et al.* [16] proposed a hybrid approach that blends a rule repository with a fuzzy Bayesian network, thereby significantly improving reliability and real-time decision quality under uncertainty. Park *et al.* [17] employed a method grounded in differential game theory to develop an automated maneuver-generation algorithm for a UAV Within Visual range.

The algorithm follows a layered decision logic and derives a score function matrix from differential game principles to pinpoint the optimal value. At the cutting edge, the most capable decision-making methods are overwhelmingly based on reinforcement learning [18]. These methods can thoroughly capture the high-dimensional characteristics of the UAV cooperative-conflict dynamics without relying on handcrafted features, thereby enabling judicious decision-making. They necessitate neither domain-specific foreknowledge nor human-derived heuristics, only a correctly specified loss function and optimizer for iterative feature learning [11]. Prevalent dynamic decision-making methods based on reinforcement learning broadly split into value-function-centric methods, such as deep Q network (DQN) and dual DQN (DDQN) [12-14], and policy-centric methods, such as actor–critic (AC) and deep deterministic policy gradient (DDPG). Zhang *et al.* [19] presented a dynamic decision-making procedure leaning on a Q-network and Nash equilibrium strategies to enhance the sample efficiency of reinforcement learning. Lillicrap *et al.* [20] introduced a deep deterministic policy gradient algorithm based on the actor–critic paradigm and extended DQN to the continuous action domain to enable direct policy learning.

Recommendation systems, a pillar of contemporary AI applications, have achieved notable triumphs in sectors such as electronic commerce, tailored learning, and intelligent transport [21-23]. Currently, the principal recommendation techniques deployed in these systems include content-based, collaborative, and hybrid algorithms [24]. In more recent times, with the emergence of advanced artificial intelligence, reinforcement learning and deep learning-driven recommendation algorithms have likewise come to the fore [25, 26].

© 2022 The Author(s). | Published by Agro Intelligence Research Press

Content-based filtering algorithms derive their suggestions primarily from a user's historical interaction footprint. In contrast, collaborative filtering (CF) algorithms exploit the behavioral records of other users to recommend relevant items to the target user. CF techniques broadly fall into two families: memory-driven and model-driven. The memory-driven branch subdivides into user-based collaborative filtering (UserCF) and item-based collaborative filtering (ItemCF) [27]. UserCF infers ratings by identifying shared behavioral patterns among users, whereas ItemCF estimates ratings based on how similar items are and which ones the user actually selected. Model-driven collaborative filtering approaches craft recommendations by fitting a predictive function. These strategies routinely model user tastes from prior interaction logs. Typical implementations encompass matrix factorization (e.g., Singular Value Decomposition, SVD [28]), latent factor models [29], and deep learning architectures that have proliferated in recent years [30]. Model-driven CF algorithms typically scale more effectively, particularly when dealing with large datasets. Their core strength is the ability to extract latent interest signals from the model, resulting in higher-fidelity recommendations.

Standard recommendation paradigms struggle to capture the sequential fingerprint of user conduct, fall short of modeling successive interdependencies between items, and are plagued by data sparsity and dependency bottlenecks [31]. The current state of the art in recommendation performance is held by reinforcement learning (RL) methodologies [32-34]. RL-based recommender approaches can be classified into classical RL and deep reinforcement learning (DRL) variants. As a form of interactive recommendation (IR), its model dynamically refreshes the suggestion policy by exchanging signals with users and harvesting their genuine responses, a scheme that mirrors actual recommendation contexts far better than fixed, static methods [35]. In parallel, because RL problems are canonically cast as Markov decision processes (MDPs), these models are intrinsically structured to model user behavior as a sequence, offering a full-fidelity depiction of sequential dynamics and capturing drifting user preferences. Moreover, the built-in exploration apparatus empowers the agent to probe the state-action landscape more comprehensively, injecting diversity into the resulting recommendations. Ultimately, these models commonly target maximizing the cumulative system reward—namely, the user's long-term feedback—as the optimization criterion for policy updates, thereby elevating enduring user contentment [36].

By encoding user preferences and fostering sustained engagement, recommendation systems enable dynamic, adaptable decision-making, a quality that aligns naturally with complex, real-time settings. Yet, despite this promise, their integration into drone maneuver decision-making remains largely uncharted territory, opening a window for the fusion of these two disciplines.

Building on the technical foundations of recommendation engines and maneuver decision logic, this work dissects their structural parallels. It proposes an original algorithm for maneuver decision-making grounded in principles of recommendation systems. Its principal contributions are:

(1) It pioneers a recommendation-system-driven maneuver decision algorithm; this groundbreaking integration opens new intellectual pathways for applying recommender systems to UAV cooperative and adversarial tasks.

(2) It assembles a recommendation environment that mirrors UAV collaborative and competitive engagements. The singular hallmark of this sandbox is an integrated architecture that unifies a variety of highly capable maneuver decision algorithms to stage multiple confrontation modes, spanning both continuous and discrete action domains. On this foundation, a discrete-action expert algorithm combining offensive and defensive postures, along with a greedy-offensive expert algorithm, is constructed as a comparative benchmark.

(3) To substantiate the viability and potency of channeling recommendation systems into the maneuver decision problem, this study builds on collaborative filtering principles. It develops an offline ensemble KNN-UserCF-based recommender for maneuver decision-making. It fuses KNN into UserCF to circumvent the burdensome computation of the user similarity matrix. It employs a Bagging ensemble strategy to aggregate diverse K-nearest neighbor sets, thereby improving recommendation stability.

© 2022 The Author(s). | Published by Agro Intelligence Research Press

(4) Confronted with the data-dependency frailties inherent to conventional offline recommendation pipelines, DRL technology is mobilized to craft and realize online maneuver decision recommenders built atop discrete-action DDQN and continuous-action DDPG, respectively. Both variants infuse prioritized experience replay into the vanilla DRL workflow to heighten sample efficiency and sculpt a dense reward signal informed by a situation assessment function. This reward design guides the UAV toward tactically superior positions in airspace to fulfill mission objectives, drastically reducing algorithmic convergence time.

Research challenges are shown in **Figure 1**.

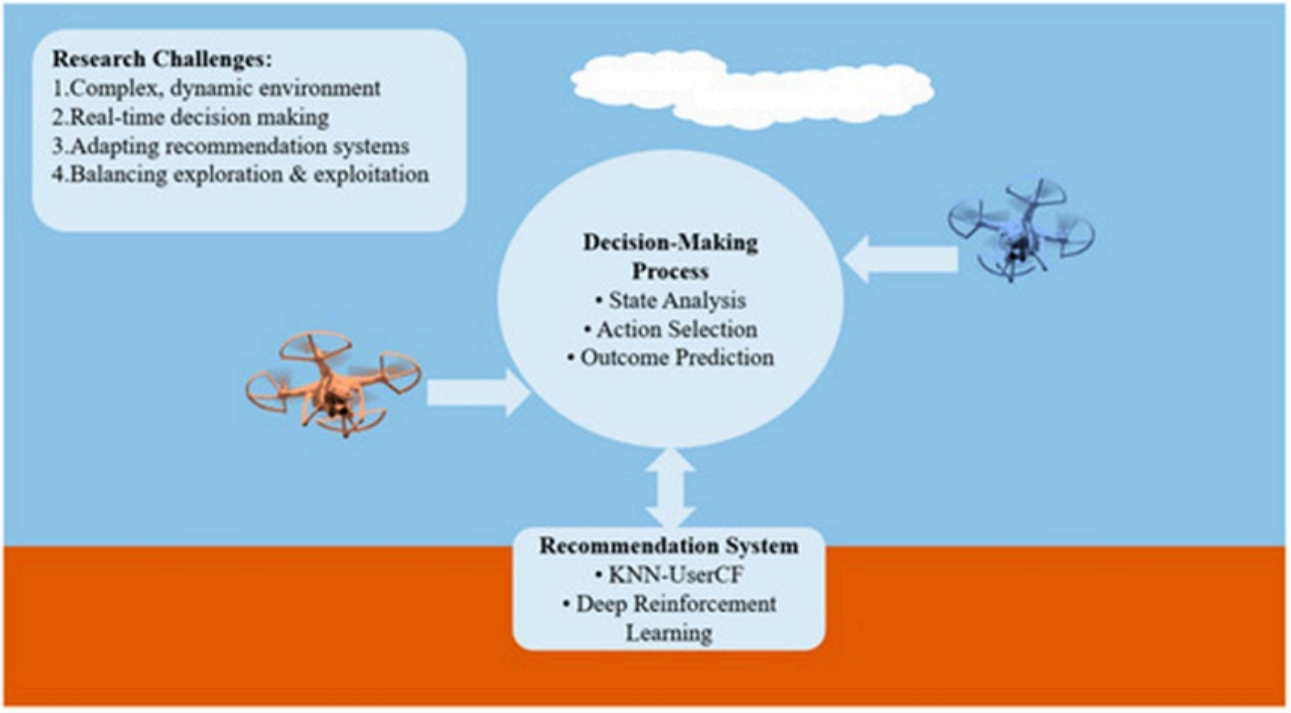

Figure 1. Research challenges and decision-making process.

The remainder of the paper is structured as follows: Section 2 sketches the integrated recommendation algorithm framework by setting up the simulation ecosystem for the UAV cooperative-countermeasure recommender. Section 3 elaborates on the KNN-UserCF recommendation model and introduces the ensemble learning module. Section 4 presents a DRL-based recommendation algorithm that builds on DDQN and DDPG, weaving in prioritized experience replay to address catastrophic forgetting caused by uniform experience sampling. Section 5 presents simulation results that highlight the win rate and robustness of the proposed autonomous maneuver decision recommender framework in UAV cooperative-countermeasure missions. Section 6 concludes.

Building the UAV Cooperative–Adversarial Simulation Environment for the Recommender System

This section describes the construction of a recommender system engineered for UAV collaborative and competitive settings. It starts from reinforcement learning-driven recommendation frameworks and covers flight dynamics and control formulations, mission success/failure evaluation criteria, and the design of a maneuver library. It further presents the integrated architecture and proposes two baseline algorithms: the offensive-only greedy expert and the combined offensive–defensive greedy expert. The environment developed here assumes idealized conditions, with no account taken of disturbances like wind, communication delays, or sensor noise.

© 2022 The Author(s). | Published by Agro Intelligence Research Press

Reinforcement-learning-based recommender system

In the reinforcement learning paradigm, both recommender systems and maneuver decision problems can be framed as Markov processes specified by a five-component tuple, {S, A, P, R, γ}, as illustrated in **Figure 2**. Compared with recommender systems, maneuver decision-making represents a Markov process operating on a much faster timescale and with higher temporal dimensionality.

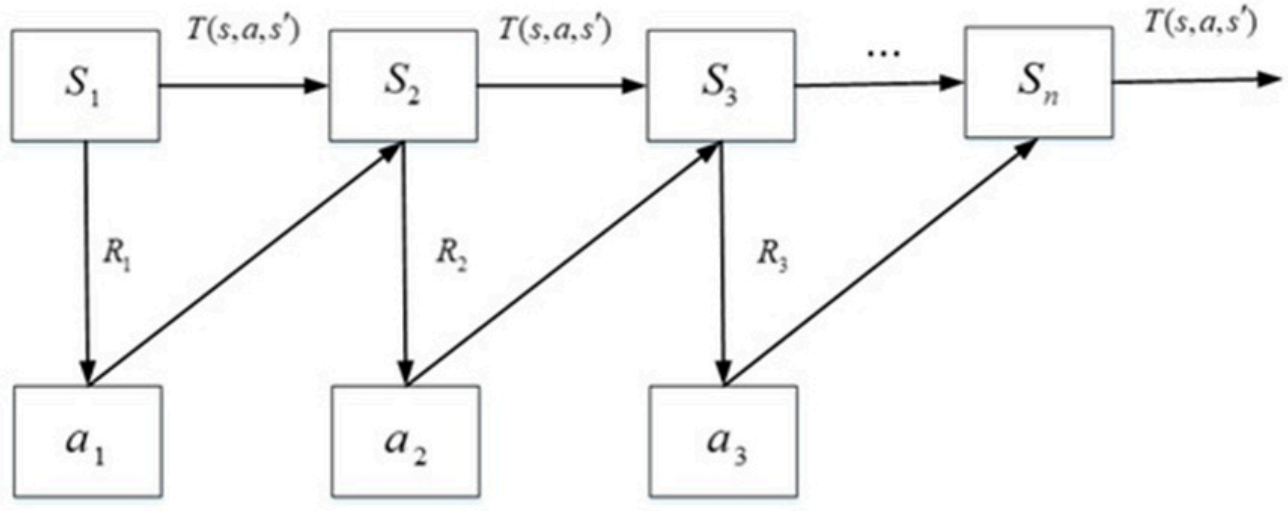

Figure 2. Schematic diagram of the Markov decision process.

Here, S = (S_1, S_2, …, S_t) denotes the state collection, capturing the situation that emerges from agent–environment interaction. Within a recommender, this corresponds to the ensemble of user states; within maneuver decision-making, it corresponds to the ensemble of UAV cooperative–adversarial states.

The quantity A = (a_1, a_2, …, a_t) denotes the action collection, i.e., the pool of choices open to the agent. For a recommender, this translates into the set of items available to suggest; for maneuver decision-making, it translates into the set of feasible maneuver commands.

The quantity P gives the state-transition kernel, describing the probability distribution of the next state conditioned on the current state–action pair.

The quantity R = (R_1, R_2, …, R_t) serves as the reward function, determined jointly by state and action, conveying the immediate signal the agent receives upon acting. In a recommender, it captures user reactions, e.g., a like or dislike; in maneuver decision-making, it captures the instantaneous assessment of the UAV's cooperative–adversarial situation, e.g., a win or a loss.

The quantity γ stands for the discount factor, balancing immediate and delayed returns. A factor closer to one emphasizes long-run payoffs, whereas a factor closer to zero stresses near-term user gratifications. Picking a suitable discount factor is a key tuning decision in reinforcement learning, as it directly molds the learned policy and value function. The discounted return can be expressed as:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (1)$$

© 2022 The Author(s). | Published by Agro Intelligence Research Press

Where G_t is the cumulative discounted payoff from time step t, and R_{t+1}, R_{t+2}… represent future rewards at successive steps.

Inside the reinforcement-learning recommender pipeline, four principal components exist, identical to those in the reinforcement-learning maneuver decision pipeline. In a recommender, state representations typically bundle item features, user profiles, and embedded encodings, with the most common formulation employing user, product, or contextual features as the user state (written as S = (S_item, S_user, …, S_context)). In maneuver decision-making, state representations typically bundle the UAV's 3-D coordinates, attitude information, and relative geometry with respect to allies and adversaries (written as S = (S_position, S_rel_pos, …, S_attitude)).

Policy optimization: In a recommender system, it seeks to refine the recommendation strategy to fit user tastes and habits better. The standard policy update rule reads:

$$\theta' = \theta + \alpha \nabla J(\theta) \quad (2)$$

Where θ represents the policy parameters, θ′ denotes the parameters after the update, α is the learning rate controlling the step size, and ∇J(θ) is the gradient of the objective with respect to the policy parameters. The same expression holds for maneuver decision-making, where the goal is to sharpen the maneuvering strategy to defeat the opponent more reliably.

Reward design: In reinforcement learning, how the reward signal is shaped directly determines the quality of the acquired behavior. Two approaches exist for specifying rewards: a bare scalar signal or a composite signal aggregating multiple observed quantities. In a recommender, the reward function often takes the form R = (u, i), denoting the reward obtained when presenting item i to user u. In maneuver decision-making, it takes the form R = (S, a), denoting the reward the UAV receives for executing action a while situated in cooperative–adversarial state S.

Environment modeling: Crafting a faithful environment is crucial for evaluating the potency of a reinforcement-learning solution. Environment modeling generally falls into three categories: offline, online, and simulated settings. In a recommender environment, modeling aims to encode user preferences, interaction histories, and other factors that may influence recommendation outcomes. A typical formulation is S′ = f(S, i), where S′ is the user's next state and f is the environment transition function triggered when item i is recommended in state S. In maneuver decision-making, an analogous formulation applies: S′ = f(S, a), where S′ is the UAV's next state and f is the environment transition function triggered when action a is taken in state S.

Within the reinforcement-learning landscape, both maneuver decision-making and recommender systems can be viewed as a recurring cycle of observe–assess–decide–act. Inherent technical symmetries link the two domains. Their components can therefore be aligned: maneuver actions map to recommended items, and user states map to UAV states. Where a recommender aims to surface the most pleasing items for users, maneuver decision-making aims to prescribe the most advantageous actions for UAVs. The mapping from the recommender system to the reinforcement-learning formulation is summarized below **(Table 1)**:

Table 1. The mapping table of the recommendation system to reinforcement learning.

Recommendation System Module	Reinforcement Learning Module
User state representation	State
Suggested item	Action

© 2022 The Author(s). | Published by Agro Intelligence Research Press

Feedback from the user	Reward signal
Recommendation strategy/algorithm	Policy
Context of recommendation	Environment
Cold-start vs long-term optimization	Exploration vs exploitation trade-off

Flight dynamics and control model development

Following the particle-dynamics modeling approach outlined by Zhou *et al.* [37], a recommender-system simulation of the UAV cooperative–adversarial environment is assembled, comprising six-degree-of-freedom flight states and three control variables. The UAV inside this world is reduced to a point mass; its state is cast as the user state, s = [x, y, z, v, γ, φ] encoding the flight quantities, while its maneuver actions are cast as the recommended content, given by $control$=[nx,nz,μ]. A schematic of these quantities appears in **Figure 3**:

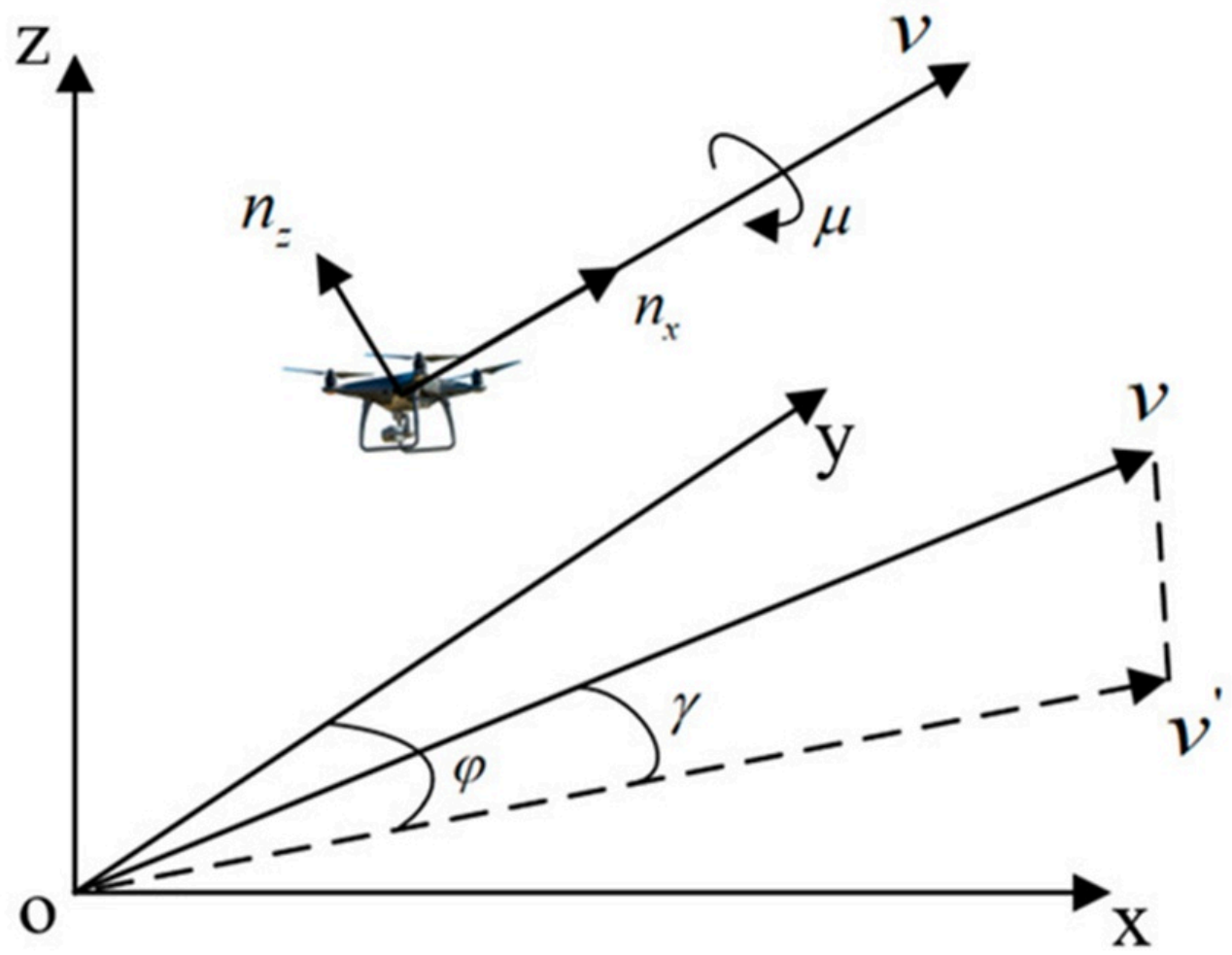

Figure 3. Schematic diagram of flight parameters.

Here, x, y, z locate the UAV in three-dimensional space, γ denotes the pitch angle (the inclination between the velocity vector and the horizontal xoy plane), and φ denotes the heading angle (the angle between the projection of the velocity vector onto the xoy plane and the positive x-axis).

© 2022 The Author(s). | Published by Agro Intelligence Research Press

Mission Win/Loss evaluation model

Using the state data of the UAVs inside the recommendation environment, we compute the relative tactical situation between the two sides, including relative range and relative aspect angles. To keep them visually distinct, our UAV is shown in red, while the opponent is shown in blue.

The variable $angleR$ measures the angle between the red-side UAV and the blue-side velocity vector. The variable $angleB$ measures the angle between the red-side agent's velocity vector and that of the blue side. The variable d measures the separation between the red and blue sides.

The conditions that follow are all treated as unavoidable loss conditions:

$$\begin{cases} angle < angle_{attack} \\ d < d_{attack} \end{cases} \quad (3)$$

Where $angleattack$ denotes the UAV's maximum firing angle (the extreme attack aspect), and d_attack denotes the UAV's maximum firing range (the attack distance). Entering this condition indicates that our UAV has lost the engagement.

Maneuver action repository

The maneuver action repository falls into two broad categories: a library of conventional tactical maneuvers and a library of elementary maneuver primitives. The elementary primitive library aligns more naturally with the maneuver decision cycle and has therefore garnered considerable attention. Within the recommender system simulation environment for UAV cooperative–adversarial engagements, a discrete maneuver repository comprising 27 distinct actions has been devised [2]. Every action corresponds to three control variables; the discrete allowable values for n_x, n_z, μ are, respectively, {−2, 0, 2}, {−2, 0, 2}, {−2, 0, 2}, while their continuous permissible intervals are (−3, 3), (−3, 9), (−π, π).

Integrated environment assembly

The recommender-system simulation environment constructed for UAV cooperative–adversarial engagements relies on an integrated architecture, as visualized in **Figure 4**:

© 2022 The Author(s). | Published by Agro Intelligence Research Press

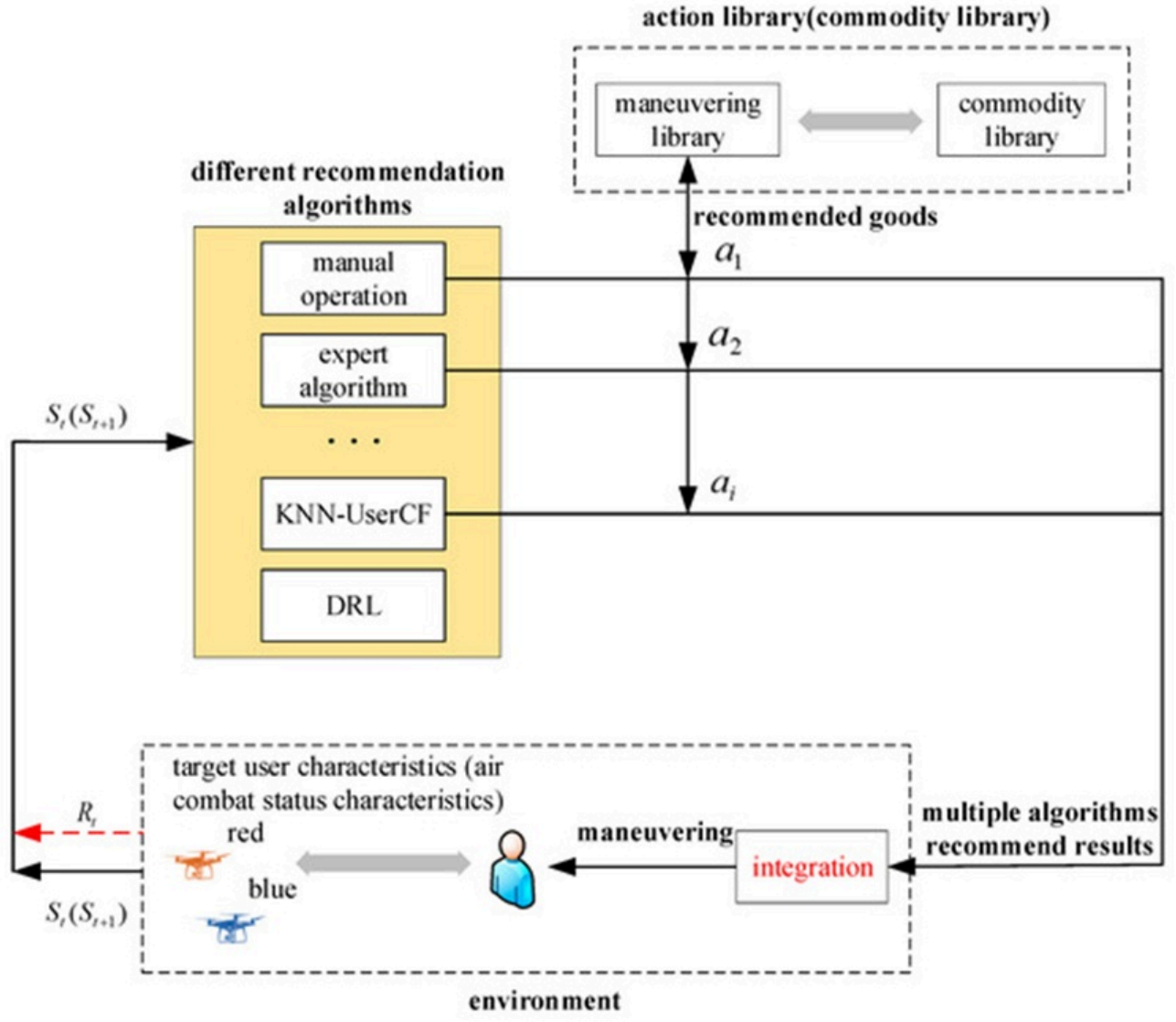

Figure 4. Integrated framework structure diagram.

The adopted integrated architecture can accommodate a diverse set of algorithms, including rule-driven policies, human-pilot inputs, and both offline and online schemes.

Suppose the integrated module is built from m weak learners; let the i-th weak learner be denoted as $learneri$, let the number of nearest neighbors each weak learner references be ki, and let the Top-N maneuver actions suggested by weak learner $learneri$ be $action_listi$=[a1,a2,…,aN]. The Bagging fusion rule for combining the m weak learners is:

$$a_j = \sum_{t=1}^{t=m} a_{t,j} \quad (4)$$

Where ai, indicates whether weak learner i puts forward action aj; ai,=1 if the action is endorsed, and ai,j=0 otherwise. The votes across all actions are aggregated, and the Top-N maneuver actions $action_list$=[a1,a2,…,aN] with the highest

© 2022 The Author(s). | Published by Agro Intelligence Research Press

vote counts are returned as the final recommendation.

Baseline algorithm formulation

Using the constructed recommender-system simulation environment for UAV cooperative–adversarial engagements, two expert algorithms are formulated as performance baselines. One is the pure-offense greedy expert, and the other is the offense–defense greedy expert. Both expert algorithms produce discrete maneuver decisions.

(1) Pure-offense greedy expert algorithm: This expert, possessing only an offensive posture, is a greedy scheme driven by range closure. The UAV iterates over all 27 maneuver types in the repository, simulates each action, and evaluates the resulting inter-agent distance. It ultimately picks and executes the action that achieves the largest reduction in separation between the two sides.

(2) Offense–defense greedy expert algorithm: Relative to the pure-offense variant, the offense–defense expert algorithm introduces a defensive component and also augments the offensive logic with a greedy criterion that seeks to shrink the aspect angle. Taking our UAV as the reference frame, the procedure first obtains the state vectors of both the opponent and our UAV, then applies a coordinate transformation to express the opponent's state in a spherical relative coordinate system centered on our UAV, thereby computing the opponent's relative position. Next, it segments our field of view into nine spatial zones, corresponding to the combinations of upper, middle, and lower bands with left, center, and right sectors. The 27 maneuver actions are classified by tangential load into acceleration, deceleration, and constant speed; by normal load into pull-up, dive, and level flight; and by roll command into left turn, right turn, and wings-level.

At every decision step, the UAV first checks whether it has encroached upon a dangerous altitude threshold. If it has, the algorithm prioritizes altitude correction to return it to the safe envelope. It then assesses the mutual tactical situation, starting with the angular advantage. The calculation reads:

$$T_{angle} = angle_R - angle_B \quad (5)$$

When $Tangle$ is negative, our side holds the angular edge; when $Tangle$ is positive, the opponent holds the angular edge; and a value of zero indicates a symmetric situation. After resolving the angular advantage, the algorithm examines the range advantage. If our side remains outside the opponent's weapons engagement zone while maintaining the angular edge, it continues refining the attack and accelerates to close the distance. If our side is already inside the engagement zone and retains the angular edge, it continues honing the attack aspect and closes at constant velocity. If our side is at an angular disadvantage but outside the engagement zone, it continues adjusting the attack angle and decelerates as it approaches. If our side suffers both an angular disadvantage and lies within the engagement zone, it continues maneuvering to escape the opponent's attack aspect and accelerates to widen the separation.

KNN-UserCF Ensemble Recommendation Algorithm

This section presents the ensemble recommendation scheme that fuses KNN with UserCF. It begins with a high-level outline of the algorithm's architecture, then provides a detailed exposition of the KNN-UserCF module, and concludes with a description of the ensemble learning module.

Architecture of the KNN-UserCF ensemble recommendation algorithm

© 2022 The Author(s). | Published by Agro Intelligence Research Press

To substantiate the feasibility and effectiveness of migrating recommender systems into the UAV maneuver decision-making problem, an ensemble learning recommendation model built on KNN-UserCF is implemented, starting from classical recommendation methodologies [38], as shown in **Figure 5**:

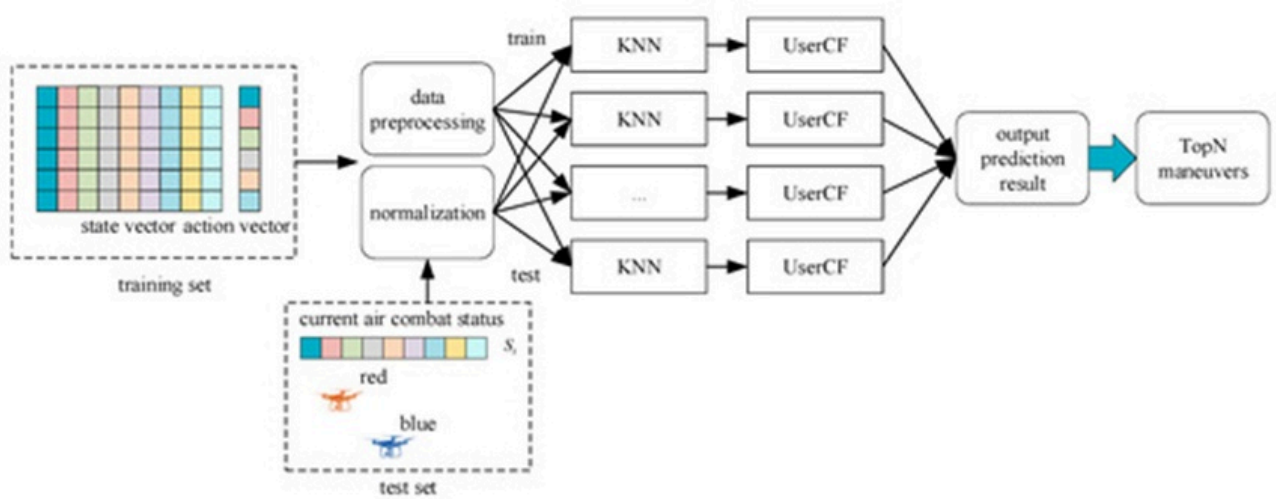

Figure 5. KNN-UserCF Integrated learning recommendation algorithm model.

Each instantaneous state is treated as a user, and each maneuver action is treated as an item. The UAV state at the current moment is first normalized and then supplied to the recommendation model; the state representation comprises 9-dimensional features. The model subsequently proposes actions by consulting multiple KNN-UserCF instances, each configured with a distinct value of K. Because the state space is enormous, directly calculating a user similarity matrix is computationally prohibitive; hence, KNN is introduced to accomplish the mapping from feature space to action space and to compute user similarity, while UserCF takes responsibility for scoring items based on that similarity. Once the outputs from the individual weak learners are collected, the Bagging ensemble strategy is used to determine the Top-N maneuver actions for recommendation.

The KNN-UserCF module

The KNN-UserCF module forms the heart of the recommendation model and is composed primarily of KNN and UserCF. KNN computes user similarity, and UserCF leverages it to recommend actions. At the current time step, the observed UAV state is first normalized to prevent large-magnitude features from overpowering the weighting, which improves robustness. Let the state feature vector be S_t = [s_1, s_2…, s_m], where m is the total number of state features and s_i is the i-th feature. The normalization of a feature can be written as:

$$s_{tnormaltze} = s_t / (s_{t\max} - s_{t\min}) \quad (6)$$

Where $sinormalize$ is the normalized value of the ith feature, simax is the maximum feasible value of s_i, and simin is the minimum feasible value. After the features are normalized, the KNN model, trained on the UAV cooperative–adversarial dataset beforehand, is used to measure the similarity between St and the feature vector St′ from another time step. Cosine similarity is routinely used in text similarity, image processing, and recommender system domains; it supplies a simple, interpretable, and effective metric of resemblance, and is therefore chosen here to quantify user similarity:

© 2022 The Author(s). | Published by Agro Intelligence Research Press

$$sim(S_t, S_{t'}) = \frac{\sum_{t=1}^{m} S_{t,t} S_{t',t}}{\sqrt{\sum_{t=1}^{m} S_{t,t}^2} \sqrt{\sum_{t=1}^{m} S_{t',t}^2}} \quad (7)$$

Where the k state users with the highest similarity to the target state user are selected as that user's k nearest neighbors, based on the actions adopted by each neighbor—i.e., the items—together with the similarity to the target user, a score is computed for every possible action. The scoring rule is:

$$action_{t,t} = \sum_{t'=0}^{k} sim(S_t, S_{t'}) \cdot action_{t',t} \quad (8)$$

Where $action_t$, is the score of action i under state St. $action_t'$ is the score assigned by state St′ to action i; if action i is taken, then $action_t'$, otherwise it is 0. Based on these scores, the Top-N actions with the highest scores are recommended to the target user.

Ensemble learning module

After producing action suggestions via the KNN-UserCF module, we treat several copies of this module as an ensemble of weak learners. Every weak learner consults a different number of neighbors, and their individual outputs are combined through an ensemble learning procedure. By applying a Bagging-style vote across the actions put forward by each weak learner, the final Top-N optimal actions are settled upon. This ensemble learning strategy materially improves the recommender system's performance, yielding a model with superior robustness and broader generalization.

Deep Reinforcement Learning Recommendation Algorithm

This section describes a recommendation framework for UAV maneuver decision-making that rests on deep reinforcement learning. It opens with an algorithm architecture designed to handle both discrete (PER-DDQN) and continuous (PER-DDPG) action spaces. It then details the prioritized experience replay mechanism and concludes by outlining the dense reward structure devised to boost learning efficiency in UAV operations.

Deep reinforcement learning framework for maneuver decision recommendation

Conventional recommendation algorithms encounter performance bottlenecks stemming from data dependencies. To bypass these limitations, adopting recommendation algorithms based on reinforcement learning can improve system performance by engaging users in real time and updating the recommendation policy online, thereby increasing the UAV mission success rate. The proposed deep reinforcement learning maneuver decision recommendation model is depicted in **Figure 6**.

© 2022 The Author(s). | Published by Agro Intelligence Research Press

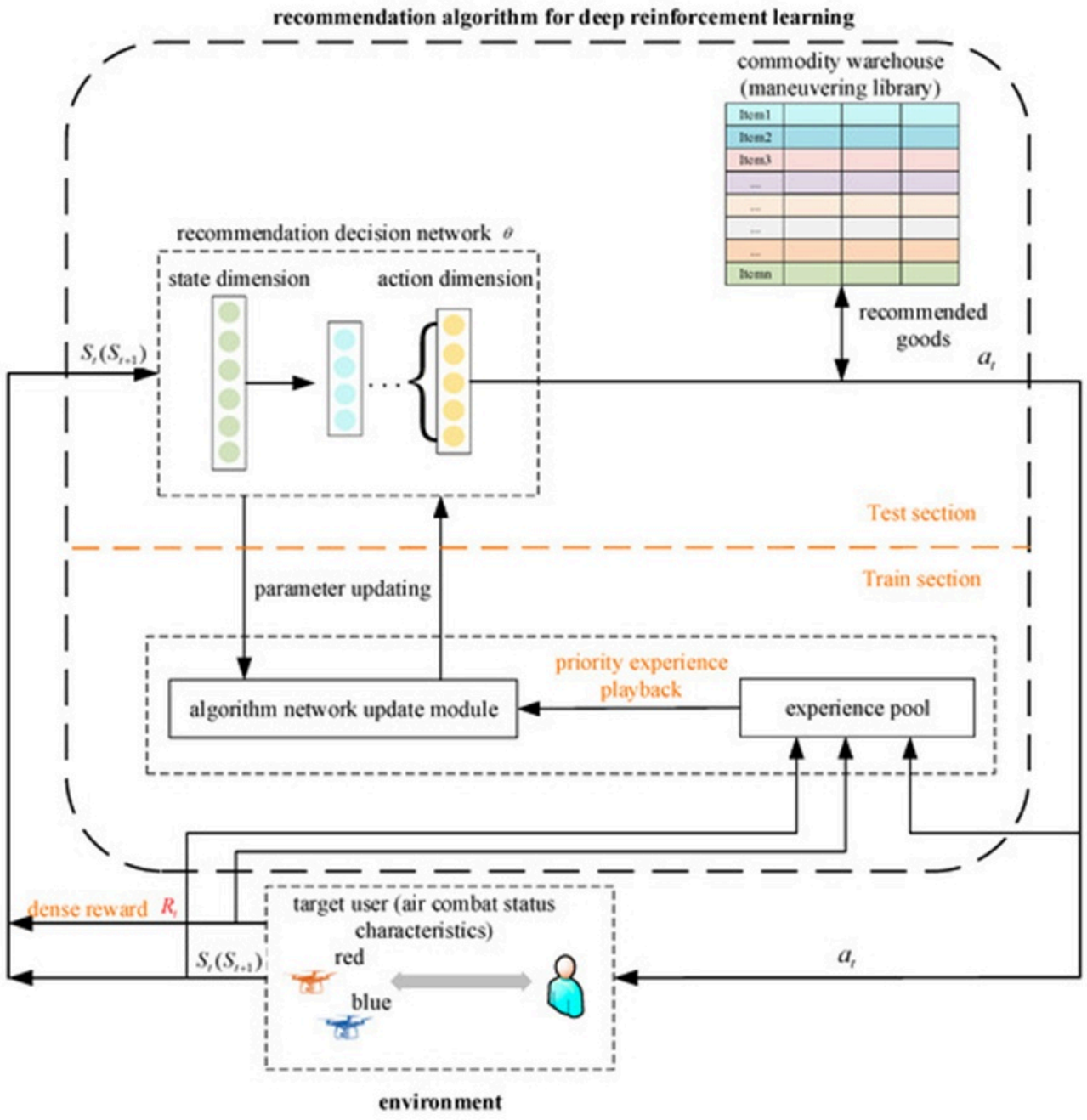

Figure 6. Diagram of a deep reinforcement learning recommendation algorithm.

Inside the model, both PER-DDQN and PER-DDPG recommendation algorithms—operating on discrete and continuous actions, respectively—are instantiated. **Table 2** below collates the merits and drawbacks of the principal existing reinforcement learning algorithms.

Table 2. Comparison of reinforcement learning algorithms for UAV tasks.

Algorithm	Advantages	Disadvantages	Applicable scenarios	Relevance to this task

© 2022 The Author(s). | Published by Agro Intelligence Research Press

DDQN [39, 40]	mitigates overestimation of Q-values	requires discretization of continuous state/action spaces	discrete decision problems, such as route planning	well-suited for UAV applications involving discrete choices (e.g., waypoint navigation or switching mission modes)
DDPG [41]	• supports continuous action spaces • efficient exploration capability • integrates actor–critic framework	• susceptible to convergence to local optima • relatively low stochasticity	continuous control problems, such as flight trajectory refinement	critical for UAV systems requiring smooth continuous control (e.g., adjusting heading angles or velocity control)
Q-Learning [42]	• straightforward and effective • easily adaptable	• poor scalability in large state–action spaces • slow learning convergence	discrete action environments, basic decision-making tasks	applicable when UAV control actions can be simplified into discrete sets
DQN [43]	simple and stable in static or low-complexity settings	• suffers from Q-value overestimation • not suitable for highly dynamic or continuous control problems	simple discrete environments with limited dynamics	inadequate for dynamic UAV scenarios due to overestimation bias and limited adaptability
A3C (Asynchronous Advantage Actor–Critic) [44]	• handles both discrete and continuous action spaces • improves learning efficiency via asynchronous updates	• computationally intensive • implementation complexity is high	dynamic and multi-task learning environments	too resource-demanding and complex for real-time UAV operations, requiring efficiency
PPO (Proximal Policy Optimization) [45]	• stable training process with strong performance robustness • applicable to both discrete and continuous actions	• high computational cost• relatively complex to implement	highly complex and dynamic environments	less suitable for UAV systems with strict real-time and computational constraints

Guided by the evaluation in the table, we selected DDQN and DDPG for our recommendation system, as they best address the specific requirements of UAV missions. DDQN, thanks to its built-in mitigation of Q-value overestimation, is well-suited to discrete decision-making contexts such as waypoint selection and mission-phase switching. DDPG, on the other hand, excels in continuous-action domains, making it well-suited for tasks that require continuous modulation, such as angular adjustment and throttle control.

View different UAV states as users, and view the maneuver action catalog as the item inventory. In each experimental episode, at each step of the recommendation cycle, the item inventory and the user state S_t are forwarded to the DRL recommendation module. The DRL module determines the item a_t with the highest value according to its decision network and recommends it to the user. The user returns feedback R_t conditioned on the suggested item—this can be a reward or a penalty, for instance, a penalty for a loss. Meanwhile, the user state transitions to the next state, and the experience buffer records the transition tuple (S_t, a_t, R_t, S_{t+1}). Afterward, the user supplies the updated state S_{t+1} to the recommendation module, which generates the next decision and recommendation, closing the loop until the UAV autonomously determines the engagement outcome. While this cycle runs, once the experience buffer

© 2022 The Author(s). | Published by Agro Intelligence Research Press

accumulates batch_size transitions, the recommendation module's network undergoes a PER-driven update, and its policy is steadily optimized toward convergence, striving to maximize the UAV's adversarial win rate.

Prioritized experience replay

Standard experience replay draws samples uniformly at random from the replay buffer, whereas prioritized experience replay selects samples in proportion to their relevance.

In prioritized experience replay, each stored transition receives a priority score, typically derived from its temporal difference (TD) error. The TD error captures the difference between the current estimate and the target value, quantifying the model's predictive inaccuracy.

Samples carrying high priority exhibit large TD errors, which flag them as more consequential for model training; a large TD error indicates that the current Q-function is significantly misaligned with the target Q-function and should undergo repeated updates. Within PER, the priority assigned to each experience is calculated as:

$$P(i) = \frac{p_t}{\sum p_t} \quad (9)$$

Where $pi=|\delta i+\varepsilon|$, with ε being a negligibly small positive number, and $\sum pi$ representing the sum of pi across all sampled experiences. This formulation ensures that experiences with zero TD error are not entirely excluded from sampling.

During training, prioritized experience replay draws samples for learning in proportion to their priority. Higher-priority samples are replayed more often, thereby amplifying the number of gradient updates applied to them and enhancing the learning signal from pivotal transitions. This steers the model's focus toward the samples most instrumental in boosting its capability.

Dense reward

In reinforcement-learning recommender systems, the construction of the reward signal is crucial to the quality of the agent's learning. Sparse rewards, though prevalent in many real-world tasks, come with disadvantages such as sluggish learning and challenging exploration. Agents need prolonged interaction to collect sufficient feedback and have difficulty identifying worthwhile states or behaviors.

Dense rewards, by contrast, supply a feedback signal at every time step, conferring notable benefits. They expedite learning and enhance the agent's ultimate competence. In deep reinforcement learning-based maneuver decision recommendation algorithms, dense rewards are particularly impactful. By obtaining user feedback immediately after each recommendation, the system can recalibrate its strategy more frequently, thereby accelerating the convergence of the decision network.

In short, implementing a dense reward mechanism in deep reinforcement learning-based recommendation systems can effectively overcome the shortcomings of sparse rewards. It equips the agent with richer and more immediate feedback, enabling more efficient learning and more accurate recommendations.

Simulation Results and Analysis

This section outlines the experimental setup and the analysis of simulation results.

Experimental setup

© 2022 The Author(s). | Published by Agro Intelligence Research Press

(1) Evaluation metrics: The evaluation metric selected is the UAV mission success rate. Let N denote the total number of engagement rounds between the red and blue sides; the success rate for each side is statistically computed as:

$$score = \begin{cases} 1, \ Red\ finish \\ 0, \ Blue\ finish \\ 0.5, \ \text{Draw} \end{cases} \quad (10)$$

From this counting procedure, the average success rate is derived as:

$$rate = \frac{\sum_{t=1}^{epoch} score_t}{epoch} \quad (11)$$

Where $scorei$ represents the outcome obtained in the ith experimental round.

(2) Initial state settings: In this study, the proposed recommendation model is placed in adversarial trials against expert algorithms. The trained KNN-UserCF ensemble recommendation algorithm governs the red side, whereas the blue side is controlled by the offense–defense greedy strategy expert algorithm. The initial conditions of the two UAVs are provided in **Table 3**.

Table 3. Initial state settings.

Name	Heading	Pitch	Speed	z	y	x
red side	[0–2π]	[0–2π]	300 m/s	5000–10,000 m	random	random
blue side	[0–2π]	[0–2π]	300 m/s	5000–10,000 m	random	random

The temporal resolution of the experiments adheres to the sampling rate of Prepar3D (P3D) (https://www.prepar3d.com/). P3D is a flight simulation environment produced by Lockheed Martin. In the experiments, the time step is fixed at 0.02 s (corresponding to a 50 Hz sampling rate), and the maximum step count per engagement round is bounded at 3000 to maintain a sensible simulation length and manageable computational load.

Examination of simulation outcomes for the KNN-UserCF ensemble maneuver decision recommender

(1) Establishing the K-nearest neighbor count and the ensemble size: A K-fold cross-validation methodology is used to independently determine both the optimal K value and the appropriate number of base learners. Throughout the experimental runs, the fold parameter k is conventionally assigned a value of 10, while the sweep for the K-nearest neighbor count extends across the interval [0, 20]. A graphical depiction of prediction accuracy as a function of K reveals that for values above 5, the model's efficacy plateaus, with accuracy oscillating around 92.6%. The inference is that once the neighbor count exceeds 5, the model reliably exhibits both elevated and steady performance.

Using the selected K-nearest neighbors parameter, each constituent weak learner is configured with exactly 10 neighbors. Following this calibration, an empirical sweep is conducted where the cardinality of the weak-learner pool is

© 2022 The Author(s). | Published by Agro Intelligence Research Press

varied from 1 through 100. During the ensemble aggregation phase of the recommender, the trend indicates that recommendation precision increases as more weak learners are incorporated. That said, the gains taper off beyond an ensemble size of 10, at which point accuracy stabilizes at around 92.5%. In weighing the compromise between computational expenditure and predictive performance, this investigation settles on an ensemble of 10 weak learners—a design choice that maintains recommendation fidelity while keeping processing burden in check.

(2) Interpretation of findings: To gauge how the volume of suggested actions influences the mission success metric, test configurations were established in which the system proposes 1, 2, or 3 candidate actions. Every configuration underwent 1000 complete simulation episodes. The policy governing action execution operates as follows: should a solitary action be put forward, that action is executed without further deliberation; should a set of multiple actions be put forward, a single member is drawn under a uniform probability distribution. The resulting data reveal that recommending a single action yields a mean success rate of 50.8%, recommending two actions nudges it to 51.0%, and presenting three actions yields a mean success rate of 50.2%.

Under the single-action recommendation regime, both the KNN-UserCF ensemble learning recommender and the expert baseline each hover around the 50% success threshold. As the pool of suggested actions grows, the success metric continues to oscillate around the 50% benchmark. What emerges from this pattern is that while the success rate is modulated by the UAV's initial spatial coordinates and orientation, causing some variation across episodes, it remains broadly centered near the halfway mark. These empirical outcomes affirm that the assembled KNN-UserCF ensemble recommendation framework is practically feasible for autonomous maneuver decision tasks and has yielded encouraging performance. Crucially, the framework condenses the original repertoire of 27 discrete maneuver categories to a maximum of three candidate actions across the autonomous decision sequence, thereby markedly improving decision throughput.

The interaction traces harvested from engagements where the expert algorithm executes an offense–defense policy serve as the training corpus for the KNN-UserCF ensemble recommender. Once trained, the resulting model is tested in simulated contests against the expert algorithm that operates under a purely offensive greedy doctrine. In this matchup, the red force is piloted by the KNN-UserCF ensemble maneuver decision recommendation algorithm, while the offensive greedy strategy expert steers the blue force. Spanning close to 1000 engagement rounds, the red force secures an average success rate of 60.8%, in contrast to the blue force's 39.2%. This outcome corroborates that the KNN-UserCF ensemble maneuver decision recommender engineered in this study successfully assimilates effective maneuvering heuristics from the engagement logs of the offense–defense expert algorithm.

Analysis of simulation outcomes for deep reinforcement learning maneuver decision recommendation

(1) Sparse-reward deep reinforcement learning recommendation system: In the sparse-reward paradigm, a scalar reward signal is disbursed to the red-side UAV strictly upon episode termination, contingent on the final environmental state, with absolutely no intermediate feedback delivered throughout the episode's unfolding.

The sparse reward specification formulated in this study adopts the following structure:

$$score = T(S_t) \quad (12)$$

© 2022 The Author(s). | Published by Agro Intelligence Research Press

$$T(S_t) = \begin{cases} T_1 \text{ if win} \\ T_2 \text{ if lose} \\ T_3 \text{ if draw} \\ T_4 \text{ otherwise} \end{cases} \quad (13)$$

Within **Eq. 13**, the constants are assigned as follows: T_1 is pegged at 1000, T_2 at −500, T_3 at 10, and T_4 at 0.

Built on this reward specification, the present paper implements both the PER-DDQN and PER-DDPG deep reinforcement learning decision-making algorithms. Upon completion of approximately 1000 training episodes, the behavioral profiles of these two algorithms under sparse reward conditions are captured and rendered in **Figure 7**:

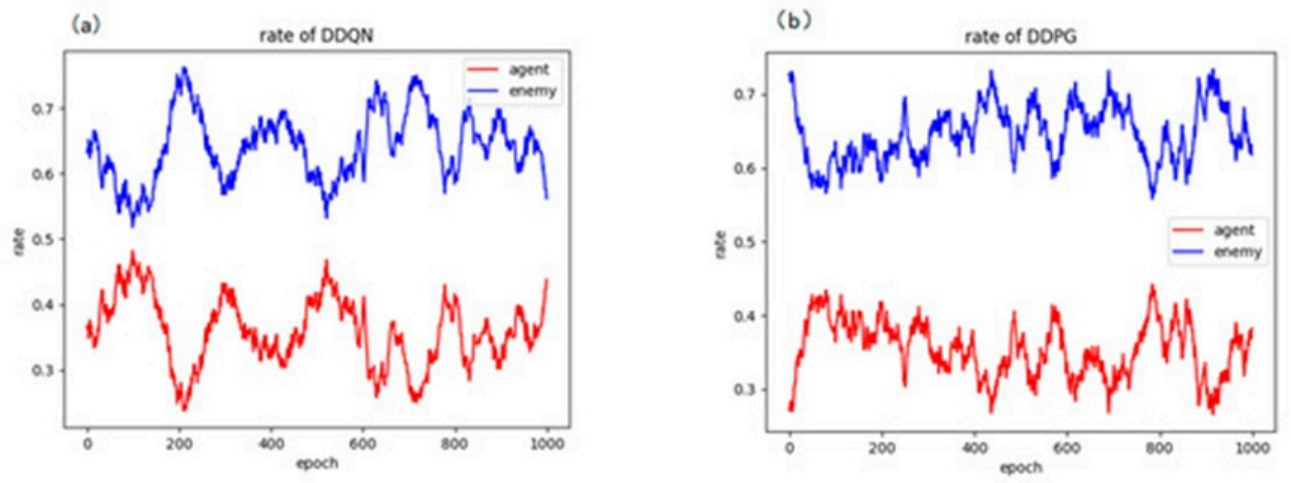

Figure 7. Evolution of the win rate for PER-DDQN and PER-DDPG when trained with sparse rewards. (a) Win rate trajectory for PER-DDQN; (b) Win rate trajectory for PER-DDPG.

A close reading of **Figure 7a** reveals that, after approximately 1000 episodes, the PER-DDQN reinforcement learning-based decision recommendation system trained on sparse rewards can achieve a maximum success rate of approximately 40% when using the expert baseline. However, its sustained success rate fluctuates around 35%.

A parallel examination of **Figure 7b** reveals that, following a comparable training investment of roughly 1000 episodes, the PER-DDPG reinforcement learning decision-making recommendation system trained on sparse rewards likewise plateaus at a success rate of about 35% relative to the expert baseline.

The empirical record indicates that, in the context of autonomous UAV maneuver decision recommendation, a deep reinforcement learning algorithm that subsists on sparse rewards struggles to extract competent maneuver policies from interactive engagements, precisely because the reward feedback harvested in cooperative–adversarial encounters is far too thinly distributed. The clear remedy is to construct a dense reward function to guide the UAV's training properly.

(2) Dense-reward maneuver decision recommendation system: In light of the difficulty that a reinforcement learning algorithm nourished by sparse rewards encounters in distilling effective strategies from engagements, a dense reward specification is architected to tutor the UAV. The crafted dense reward is composed predominantly of three components: an angular reward, a range-based reward, and a situation assessment reward. The algebraic formulation of the dense reward is presented below:

© 2022 The Author(s). | Published by Agro Intelligence Research Press

$$score = T(S_t) + T(angle_reward) + T(angle_punish) + T(dis) + T(score1) - M \quad (14)$$

In this expression, $T(St)$ captures the terminal bonus or penalty dispensed upon episode conclusion, with its numerical assignments mirroring those appearing in **Eq. 14**; $T(angle_reward)$ quantifies the extent to which the blue force sits inside the red force's aspect cone—when the subtended angle surpasses π/6 the contribution is zeroed, and when it falls below π/6 the contribution equals $\cos(angle_R)$, where $angle_R$ measures the angular separation between the blue-side UAV and the red-side velocity orientation; $T(angle_punish)$ analogously quantifies the degree to which the red force falls within the blue force's aspect envelope—angles above π/6 yield zero, while angles below π/6 yield $-\cos(angle_R)$, with $angleB$ gauging the angular offset between the red-side UAV and the blue-side velocity orientation; T(dis) furnishes a proximity-based reward between the opposing sides, taking the value 0.5 whenever the red force advances to within a predetermined threshold distance of the blue force, and defaulting to 0 in all remaining circumstances; T(score1) supplies the situation assessment reward component, with Guo *et al.* [13] resorting to the sliding-window-based CRITIC-G1 methodology to derive this continuous reward signal, which occupies the normalized interval 0–1; and M designates the lowest admissible situation assessment reward value in the course of combat, fixed at 0.38.

Capitalizing on the dense reward function defined in **Eq. 14**, this paper introduces both the PER-DDQN and PER-DDPG maneuver decision recommendation systems. Following a training regimen of roughly 1000 episodes, the performance characteristics of the two deep reinforcement learning algorithms under dense reward conditions are logged and displayed in **Figure 8**:

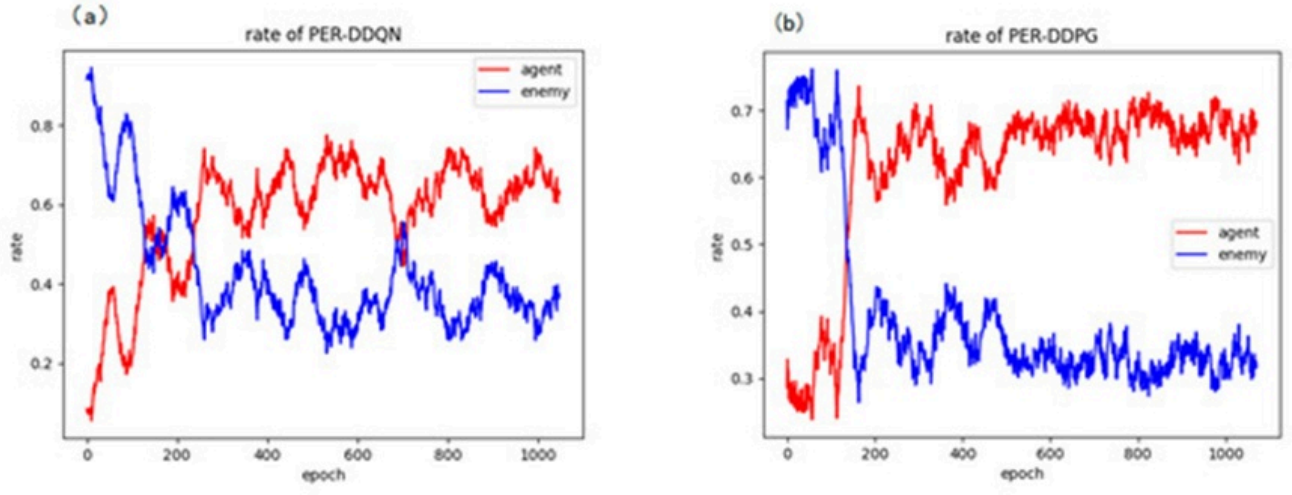

Figure 8. Evolution of the win rate for PER-DDQN and PER-DDPG when trained with dense rewards: (a) Win rate trajectory for PER-DDQN, and (b) Win rate trajectory for PER-DDPG.

Figure 8a shows that the PER-DDQN maneuver decision recommendation system, trained on dense rewards, achieves a win rate of approximately 63% in matchups against expert algorithms. This mark demonstrates substantially more efficient absorption of effective maneuvering tactics than its counterpart trained on sparse feedback.

The picture painted by **Figure 8b** indicates that, after roughly 1000 training iterations, the PER-DDPG maneuver decision recommendation system, fueled by dense rewards, achieves a win rate of nearly 69% against expert algorithms, thereby surpassing the dense-reward PER-DDQN variant. To summarize, while the PER-DDQN reinforcement learning-based maneuver decision recommendation system remains bound to the proposition of discrete action choices, the PER-DDPG recommendation algorithm distinguishes itself by recommending continuous maneuvering actions (products). It is precisely this capability that unlocks finer-grained actuation and superior adaptivity within the complex theater of air combat.

© 2022 The Author(s). | Published by Agro Intelligence Research Press

To place the efficiency and overall competence of the proposed PER-DDPG and PER-DDQN algorithms in context against traditional reinforcement learning paradigms, a rigorous head-to-head evaluation was executed. **Table 4** catalogues both the convergence horizon (operationally defined as the tally of training iterations required to reach a plateau in reward) and the mission success percentages for various algorithmic approaches when pitted against the offense–defense greedy strategy expert.

Table 4. Comparison of convergence time and mission success rates among different methods.

Algorithm	Q-Learning	DQN	PER-DDQN	PER-DDPG
Success rate	30%	45%	63%	69%
Convergence time(iterations)	5000	3000	1000	1000

As **Table 4** makes evident, PER-DDPG builds on DDPG's formidable ability to handle continuous-action domains and augments it with prioritized experience replay, so that critical transition samples encountered during learning are revisited before less informative ones, thereby compressing the convergence timeline. DQN, on the other hand, is hampered by its Q-learning lineage and converges more slowly in high-dimensional state spaces or complex mission profiles. Among the tested algorithms, PER-DDPG posts the strongest showing, achieving a mission completion rate reaching 69%. Compared to Q-Learning, PER-DDPG reduces the convergence time by 80%.

Figure 9 portrays the nascent training behavior of the red-side UAV when governed by the proposed PER-DDPG maneuver decision recommendation system conditioned on dense rewards.

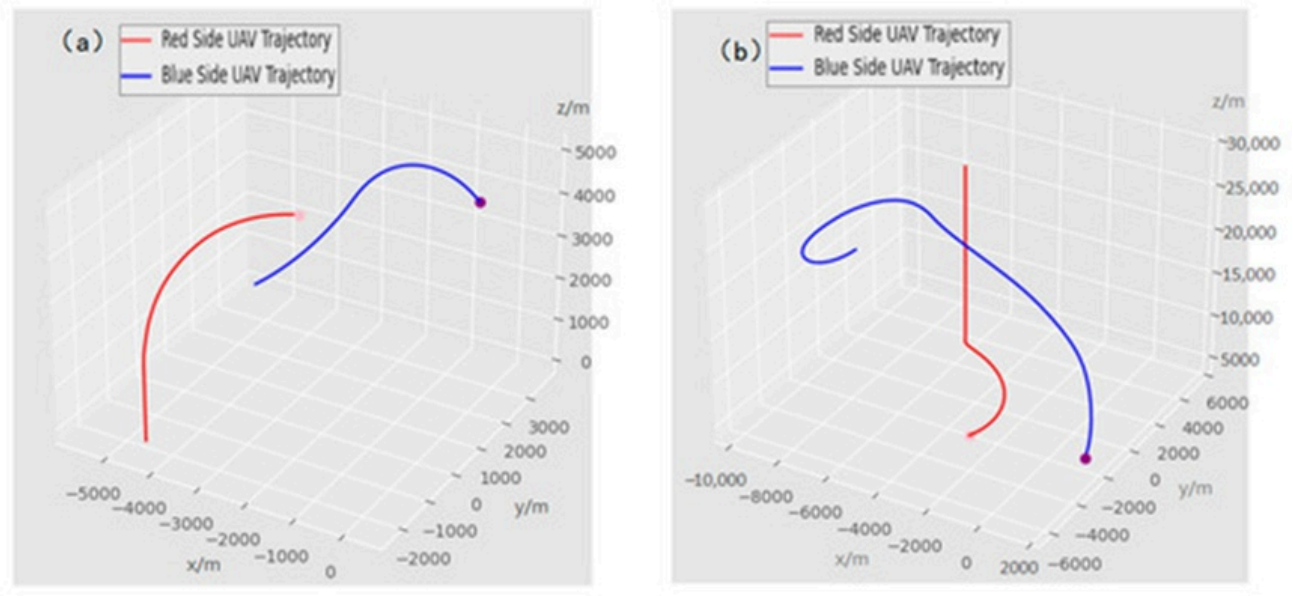

Figure 9. Defeat trajectories of the red-side UAV produced by the PER-DDPG dense-reward maneuver decision recommendation system during early training: (a) Trajectory 1, and (b) Trajectory 2.

During the earliest training epochs, the PER-DDPG recommendation system is still operating in a largely exploratory mode and has not yet crystallized an effective policy for suggesting maneuver actions. In **Figure 9a**, the actions recommended by the recommendation system for the red side temporarily put it in an advantageous geometry. The blue side opened the engagement by hunting for the red side, which merely cruised straight ahead. Ultimately, the blue side dove, realigned its attitude, and gave chase. In the closing moments, the PER-DDPG recommendation algorithm advised

© 2022 The Author(s). | Published by Agro Intelligence Research Press

the red side to fly directly downward and forward, culminating in a ground collision before the blue side satisfied the termination criteria.

In **Figure 9b**, after a modest amount of training, the recommendation system had internalized the self-destructive behavior while simultaneously beginning to piece together rudimentary offensive maneuvers. At the engagement's opening, the recommendation system prescribed actions that prompted the red side to mount an attack on the blue side, and the blue side answered by incrementally turning right to counter-engage. Neither combatant, however, satisfied the mission termination thresholds. The recommendation system then steered the red side into a straight vertical climb, betraying no inclination to seize a positional edge. The blue side closed range, spiraled up around the red side, and strove to meet the mission completion prerequisites. In the final reckoning, after the red side had traversed nearly 30,000 meters, the blue side ran it down and concluded the mission.

Figure 10 captures the matured behavior of the red-side UAV under the proposed PER-DDPG dense-reward maneuver decision recommendation system after extended training.

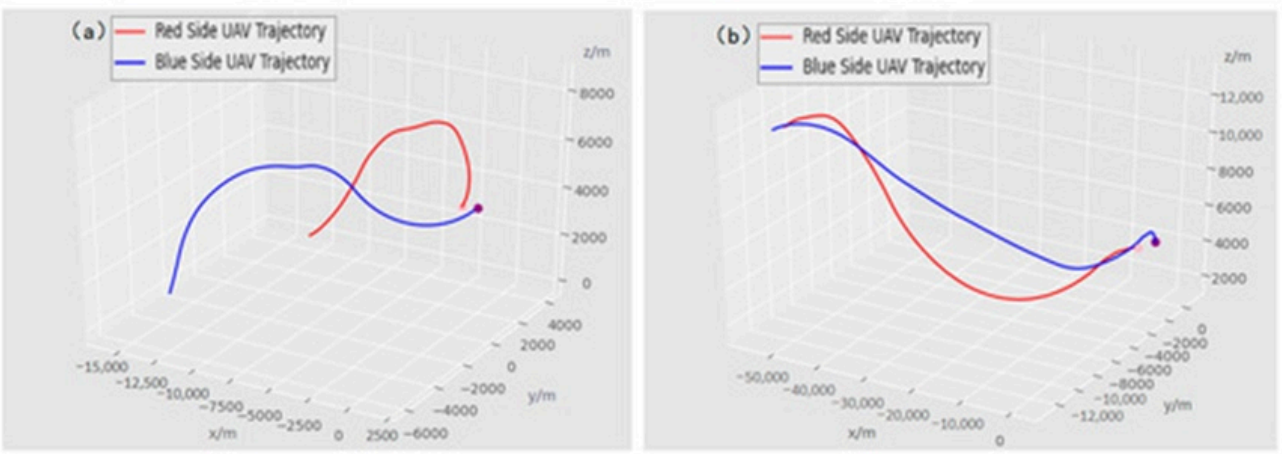

Figure 10. Victory trajectories of the red-side UAV produced by the PER-DDPG dense-reward maneuver decision recommendation system during later training: (a) Trajectory 1, and (b) Trajectory 2.

At this more advanced training stage, the scenario in **Figure 10a** begins with the blue side enjoying a more favorable angular relationship; the red side's pitch is oriented upward while the blue side's pitch points downward. Guided by the PER-DDPG recommendation algorithm, the red side executed a roll to try to narrow the gap, while the blue side actively pulled up to converge on the red side. After the initial merge, neither party triggered the mission completion condition. The red side then steadily climbed in a bid to re-engage. Still, the blue side accelerated into a dive—trading its altitude buffer for a substantial energy surplus—and crossed into the red side's forward sector at a disadvantageous angle. With its velocity now excessive, a timely pull-up became unfeasible, and it met the ground.

In **Figure 10b**, the duel commenced with the red side positioned ahead of the blue side and the blue side holding the angular upper hand. The blue side subsequently fine-tuned its orientation and bore down on the red side. The recommendation algorithm instructed the red side to dive to shake off the blue side. After the disengagement, the red side pulled into a climb to chase a height advantage, while the blue side, having fallen short of the mission completion condition on the first pass, merged with the red side. Post-climb, the red side possessed the altitude edge, and the blue side, likewise unable to close out the mission, drifted ahead of the red side at an unfavorable angle. The blue side then dove to break away, but the red side swiftly adjusted its attitude, dove, and ran the blue side down, satisfying the mission completion condition and ultimately destroying the blue side to claim the win.

© 2022 The Author(s). | Published by Agro Intelligence Research Press

Here, we scrutinize the behavior of the maneuver decision recommendation system anchored in dense rewards. The dense reward formulation was deliberately chosen for its ability to quicken skill acquisition and preserve stable gradient flow, both of which translate into markedly better training throughput.

It is nonetheless important to register that, for all its advantages, the dense reward paradigm carries certain liabilities. Chief among them is a tendency toward overfitting, particularly when the agent becomes excessively attuned to small environmental fluctuations. There is also the hazard that the reward signal may grow overly reliant on idiosyncratic features of the environment, potentially shrinking the solution manifold.

Despite these reservations, we maintain that the dense reward formulation was the most suitable design choice for the present problem, given its efficiency and the comparatively stationary nature of the operational environment.

(3) Comparison of deep reinforcement learning against the conventional recommendation algorithm: The KNN-UserCF ensemble learning maneuver decision recommendation algorithm and the deep reinforcement learning maneuver decision recommendation system trained on dense rewards are each evaluated in contests against the expert algorithm, with the comparative success rate curves collected in **Figure 11**:

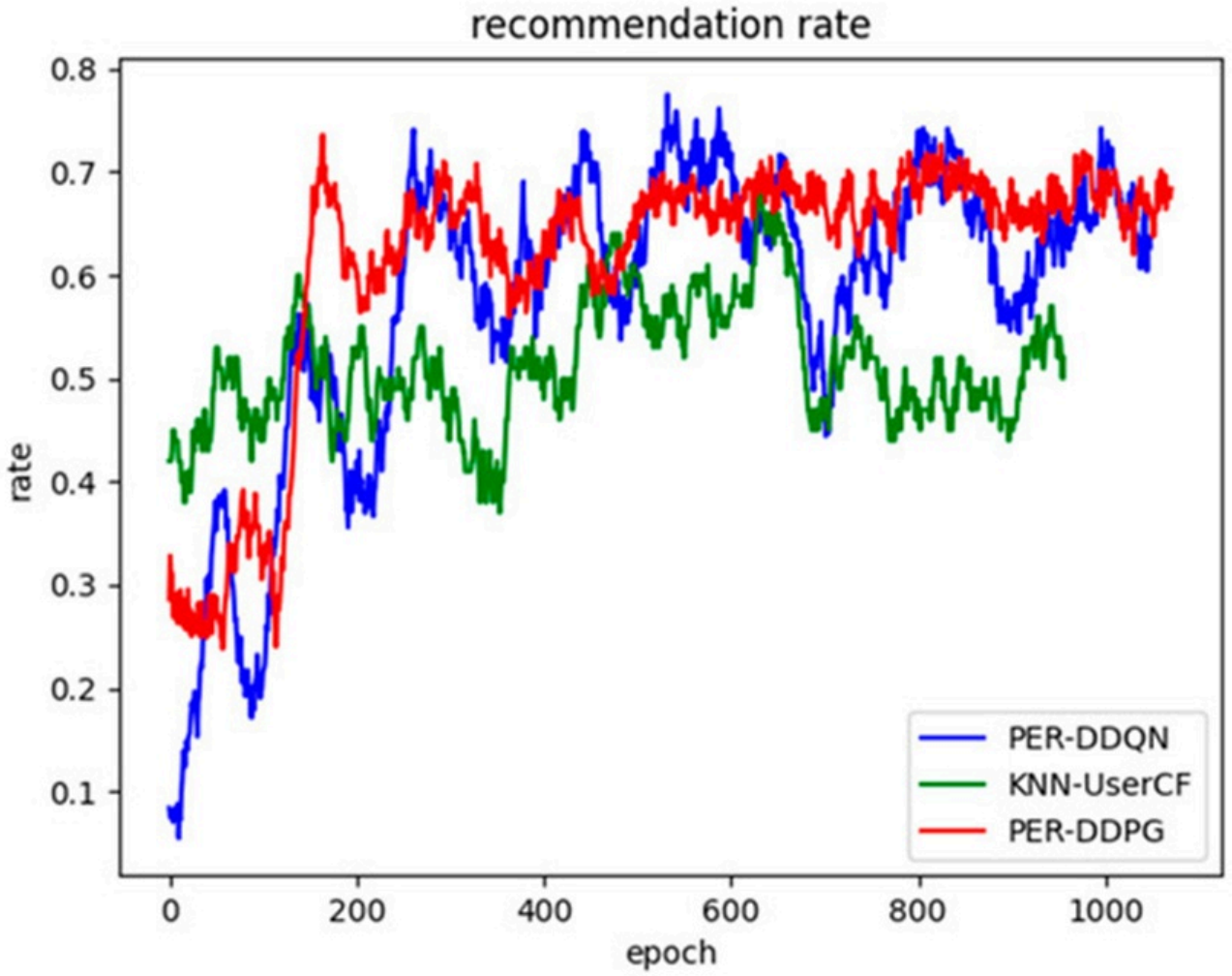

Figure 11. Comparative chart of winning rates across the three maneuver decision recommendation algorithms.

© 2022 The Author(s). | Published by Agro Intelligence Research Press

During the early phase of deep reinforcement learning training, the success rate started low but steadily increased as the episode count rose, eventually exceeding the 50% threshold against the expert algorithm. Over the same span, the conventional maneuver decision recommendation algorithm's success rate remained stuck at approximately 50%. A principal strength of the proposed framework lies in its ability to blend deep reinforcement learning with prioritized experience replay, thereby improving sample efficiency and driving the emergence of optimal decision policies across a range of scenarios. Beyond this, the integrated architecture accommodates contributions from multiple human specialists and advanced algorithmic modules, enabling a collaborative decision-making process that flexibly responds to dynamic, real-time conditions.

Between the two deep reinforcement learning recommendation variants, the PER-DDPG algorithm—which recommends actions drawn from a continuous manifold—produces a more stable success rate trajectory than PER-DDQN, whose recommendations are restricted to a discrete set. The continuous-action products, i.e., the maneuver commands, are more finely differentiated, and the associated success rate is correspondingly higher.

For real-time platforms, the time step directly governs both response latency and decision quality. Selecting an appropriate time step is therefore essential to achieving real-time operability. In keeping with the P3D configuration, this study fixed the time step at 0.02 seconds and imposed a maximum per-episode step budget of 3000. Benchmarked against these settings, PER-DDQN shows slower responsiveness in real-time decision-making tasks, eroding decision quality. PER-DDPG, in contrast, demonstrated stronger stability when operating in continuous action spaces.

Conclusion

This paper represents the first effort to transplant recommendation system methodologies into the arena of autonomous UAV maneuver decision-making, capitalizing on an integrated framework that unites a diverse array of algorithms to reproduce UAV cooperative and adversarial scenarios. Through exhaustive computer-based experimentation, the study confirms both the viability and the effectiveness of this framework and the accompanying recommendation algorithms in resolving maneuver decision challenges. The findings further indicate that a maneuver decision recommendation algorithm fortified with deep reinforcement learning—specifically through dense reward signals and prioritized experience replay—achieves a notably high success rate. Despite the advances reported here, the framework is not without its constraints. The substantial computational footprint of deep reinforcement learning complicates real-time deployment, and the system's dependency on high-quality data leaves it susceptible to noise and gaps. While simulation results are strong, additional validation in physical, real-world settings is indispensable to establishing robustness and adaptability. Future work should focus on improving the generalization capability of the recommendation system across a wide range of UAV mission contexts and on investigating data-robustness strategies to ensure reliable performance during real-world fielding.

Acknowledgements

None

Conflict of interest

None

Financial support

This research was funded by the National Natural Science Foundation of China through the following project number: GKJJ22012101.

© 2022 The Author(s). | Published by Agro Intelligence Research Press

Ethics statement

Current research is limited to the field of Artificial Intelligence (AI), which is beneficial for advancing technology, improving efficiency across industries, and fostering innovative solutions to complex problems. This research does not pose a threat to public health or national security. The authors acknowledge the dual-use potential of AI research and confirm that all necessary precautions have been taken to prevent misuse. As an ethical responsibility, the authors strictly adhere to relevant national and international laws concerning Dual Use Research of Concern (DURC). The authors advocate for responsible deployment, ethical considerations, regulatory compliance, and transparent reporting to mitigate misuse risks and foster beneficial outcomes.

Received: 05 Jun 2021 Revised: 10 Aug 2021 Accepted: 26 Aug 2021
Published online: 25 May 2022

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at https://creativecommons.org/licenses/by-nc-sa/4.0/.

References

1. Huang Y, Feng B, Tian A, Dong P, Yu S, Zhang H. An efficient differentiated routing scheme for Meo/Leo-Based multi-layer satellite networks. IEEE Trans Netw Sci Eng. 2024;11:1041–2024.

2. Wang Y, Ren T, Fan Z. Unmanned aerial vehicle air combat maneuver decision-making based on guided Minimax-DDQN. Comput Appl. 2023;43:2636–43.

3. Wei YJ, Zhang HP, Huang CQ. Maneuver Decision-Making For Autonomous Air Combat Through Curriculum Learning And Reinforcement Learning with Sparse Rewards. arXiv 2023, arXiv:2302.05838.

4. Yin S, Kang Y, Zhao Y, XueJ. Air Combat Maneuver Decision Based on Deep Reinforcement Learning and Game Theory. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Hefei, China; 2022. p. 6939–43.

5. Hu J, Wang L, Hu T, Guo C, Wang Y. Autonomous maneuver decision making of dual-UAV cooperative air combat based on deep reinforcement learning. Electronics. 2022;11:467.

6. Dong Y, Ai J. Decision making in autonomous air combat: Review and Prospects. Acta Aeronaut Astronaut Sin. 2020;41:724264.

7. Fu W, Wang H, Gao S. UAV Decision-Making Expert System in Dynamic Environment Based on Heuristic Algorithm. Beijing Univ Aeronaut Astronaut J. 2015;41:1994–9.

8. Fu L, Wang X. Research on differential game modeling for close-range air combat of unmanned combat aircraft. Ordnance J. 2012;33:1210–6.

9. Li K, Zhang K, Zhang Z, Liu Z, Hua S, He J. A UAV maneuver decision-making algorithm for autonomous airdrop based on deep reinforcement learning. Sensors. 2021;21:2233.

10. Huang C, Xie Q. Cooperative multi-objective attack decision-making method based on genetic algorithm. Fire Control Command Control. 2004;29:4–8.

11. Xie J, Yang Q, Dai S, Wang W, Zhang J. UAV maneuvering decision research based on enhanced genetic algorithms. J Northwestern PolytechUniv. 2020;6:38.

12. Vicsek T. Universal Patterns of Collective Motion from Minimal Models of Flocking. In Proceedings of the 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, Venice, Italy; 2008. p. 3–11.

13. Guo H, Xu H-J, Gu X.-D, Liu D-Y. Air combat decision-making for cooperative multiple target attack based on improved particle swarm algorithm. Fire Control Command Control. 2011;36:49–51+55.

14. Li SY, Chen M, Wang YH, Wu QX. Air combat decision-making of multiple UCAVs based on constraint strategy games. Def Technol. 2022;18:368–83.

15. Li S, Chen M,Wang Y, Wu Q. A fast algorithm to solve large-scale matrix games based on dimensionality reduction and its application in multiple unmanned combat air vehicles attack-defense decision-making. Inf Sci. 2022;594:305–21.

16. Geng WX, Kong FE, Ma DQ. Study on tactical decision of UAV medium range air combat. In Proceedings of the 26th Chinese Control and Decision Conference, Changsha, China; 2014. p. 135–9.

17. Park H, Lee BY, Tahk MJ, Yoo DW. Differential game based air combat maneuver generation using scoring function matrix. Int J Aeronaut Space Sci. 2016;17: 204–13.

18. Zhao Z, Wan Y, Chen Y. Deep reinforcement learning-driven collaborative rounding-up for multiple unmanned aerial vehicles in obstacle environments. Drones. 2024;8:464.

19. Zhang Q, Yang R, Yu L, Zhang T, Zuo J. BVR air combat maneuvering decision by using Q-network reinforcement learning. J Air Force Eng Univ (Nat Sci Ed). 2018;19:8-14.

20. Lillicrap TP. Continuous control with deep reinforcement learning. arXiv 2015, arXiv:1509.02971.

21. Li F, Li Q, Wang X. Thinking on the complexity of air combat under the development trend of equipment technology. Mil Dig. 2024;1:31-3.

22. Nguyen LV. OurSCARA: Awareness-based recommendation services for sustainable tourism. World. 2024;5:471-82.

23. Pradeep N, Mangalore KKR, Rajpal B, Prasad N, Shastri R. Content based movie recommendation system. Int J Res Ind Eng. 2020;9:337-48.

24. Shan J. Research on content-based personalized recommendation system [PhD thesis]. Changchun (China): Northeast Normal University; 2015. p. 3-5.

25. Bachiri K, Yahyaouy A, Gualous H, Malek M, Bennani Y, Makany P, et al. Multi-agent DDPG based electric vehicles charging station recommendation. Energies. 2023;16:6067.

26. Zhou Q. A novel movies recommendation algorithm based on reinforcement learning with DDPG policy. Int J Intell Comput Cybern. 2020;13:67-79.

27. Wang Y, Zheng Y, Xu J. Personalized recommendation system for library based on hybrid algorithm. Comput Inf Technol. 2023;31:39–42+50.

28. Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer. 2009;42:30-7.

© 2022 The Author(s). | Published by Agro Intelligence Research Press

29. Rendle S, Schmidt-Thieme L. Factorization models for collaborative filtering. ACM Comput Surv. 2010;42:1-23.

30. Zhang S, Yao L, Sun A, Tay Y. Deep learning based recommender system: A survey and new perspectives. ACM Comput Surv. 2019;52:1-38.

31. Feng B, Tian A, Yu S, Li J, Zhou H, Zhang H. Efficient cache consistency management for transient IoT data in content-centric networking. IEEE Internet Things J. 2022;9:12931-44.

32. Zhao K, Liu S, Cai Q, Zhao X, Liu Z, Zheng D, et al. KuaiSim: A comprehensive simulator for recommender systems. Adv Neural Inf Process Syst. 2023;36:44880-97.

33. Zhao X, Xia L, Zou L, Liu H, Yin D, Tang J. Whole-chain recommendations. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management; Galway, Ireland; 2020. p. 1883-91.

34. Zhao X, Gu C, Zhang H, Yang X, Liu X, Tang J, et al. DEAR: Deep reinforcement learning for online advertising impression in recommender systems. Proc AAAI Conf Artif Intell. 2021;35:750-8.

35. Afsar MM, Crump T, Far B. Reinforcement learning based recommender systems: A survey. ACM Comput Surv. 2022;55:1-38.

36. Chu WT, Tsai YL. A hybrid recommender system considering visual information for predicting favorite restaurants. World Wide Web. 2017;20:1313-31.

37. Zhou S, Shi Y, Yang W, Wang J, Gao L, Gao Y. Multi-aircraft cooperative air combat maneuver decision-making based on Cook-Seiford group decision-making algorithm. Command Control Simul. 2023;45:44-51.

38. Feng B, Huang Y, Tian A, Wang H, Zhou H, Yu S, et al. DR-SDSN: An elastic differentiated routing framework for software-defined satellite networks. IEEE Wirel Commun. 2022;29:86-2022.

39. Zhao J, Gan Z, Liang J, Wang C, Yue K, Li W, et al. Path planning research of a UAV base station searching for disaster victims' location information based on deep reinforcement learning. Entropy. 2022;24:1767.

40. Bao T, Syed A, Kennedy WS, Kantarcı ME. Sustainable task offloading in secure UAV-assisted smart farm networks: A multi-agent DRL with action mask approach. arXiv. 2024.

41. Yan C, Xiang X, Wang C. Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments. J Intell Robot Syst. 2020;98:297-309.

42. Yijing Z, Zheng Z, Xiaoyi Z, Yang L. Q learning algorithm based UAV path learning and obstacle avoidence approach. In: Proceedings of the 2017 36th Chinese Control Conference; Dalian, China; 2017. p. 3397-402.

43. van Hasselt H, Guez A, Silver D. Deep reinforcement learning with Double Q-learning. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16); Phoenix, AZ, USA; 2016. p. 1-7.

44. Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, et al. Asynchronous Methods for Deep Reinforcement Learning. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA; 2016. p. 1928–37.

45. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv. 2017;arXiv:1707.06347.

© 2022 The Author(s). | Published by Agro Intelligence Research Press