drones

Article

Optimized Dynamic Deployment of UAVs in Maritime Networks with Route Prediction

Yanli Xu *,† **and Yalan Shi** †

School of Information Engineering, Shanghai Maritime University, Shanghai 201306, China; 202330310165@shmtu.edu.cn

* Correspondence: ylxu@shmtu.edu.cn

† These authors contributed equally to this work.

Abstract: The limited coverage of terrestrial base stations and the limited transmission distance and onboard resources of satellite communications make it difficult to ensure the quality of communication services for marine users by relying only on satellites and terrestrial base stations. In contrast, UAVs, as flexible mobile communication nodes, have the capacity for dynamic deployment and real-time adjustment. They can effectively make up for the communication blind spots of traditional satellites and ground base stations in the marine environment, especially in the vast and unpredictable marine environment. Considering the mobility of maritime users, one can effectively reduce the communication delay and optimize the deployment scheme of UAVs by predicting their sailing trajectories in advance, thus enhancing the communication service quality. Therefore, this paper proposes a communication coverage model based on mobile user route prediction and a UAV dynamic deployment algorithm (RUDD). It aims to optimize the coverage efficiency of the maritime communication network, minimize the communication delay, and effectively reduce the energy consumption of UAVs. In this algorithm, the RUDD algorithm employs a modified Long Short-Term Memory (LSTM) network to predict the maritime user's trajectory, utilizing its strengths in processing time-series data to provide accurate predictions. The prediction results are then used to guide the Proximal Policy Optimization (PPO) algorithm for the dynamic deployment of UAVs. The PPO algorithm can optimize the deployment strategy in dynamic environments, improve communication coverage, and reduce energy consumption. Simulation results show that the proposed algorithm can complement the existing satellite and terrestrial networks well in terms of coverage, with a communication coverage rate of more than 95%, which significantly improves the communication quality of marine users in areas far from land and beyond the reach of traditional networks, and enhances network reliability and user experience.

Keywords: dynamic deployment of UAVs; communication coverage; maritime communication; route prediction

Citation: Xu, Y.; Shi, Y. Optimized Dynamic Deployment of UAVs in Maritime Networks with Route Prediction. *Drones* **2024**, *8*, 759. https://doi.org/10.3390/drones8120759

Academic Editors: Sai Huang, Ray E. Sheriff, Weiwei Jiang, Yafeng Zhan and Zhiyong Feng

Received: 23 October 2024
Revised: 11 December 2024
Accepted: 13 December 2024
Published: 16 December 2024

1. Introduction

In recent years, with the rapid growth of China's ships, offshore platforms, buoys, and other marine activities and facilities, the demand for low-latency, high-reliability communications at sea has become increasingly strong [1–3]. Satellite networks, as an important means of communication, can provide a wide range of network connectivity for marine users, which to a certain extent meets the needs of maritime communication. However, satellite signals are affected by weather, ocean reflections, and multi-path interference, which can easily lead to unstable signal quality and thus affect the communication experience. Meanwhile, due to the long signal transmission distance of satellite communication and the limited satellite on-board payload (e.g., antenna, power, etc.), these factors lead to significant challenges in terms of transmission rate, latency, and reliability of satellite communication in the marine environment [4,5]. On the other hand, although terrestrial base stations can provide high-speed networks for some sea areas, their coverage is extremely limited. In particular in open sea areas and areas far from land, the coastline of marine areas is long and variable, and terrestrial base stations cannot provide continuous coverage. Meanwhile, for high-speed moving targets such as ships and offshore platforms, the fixed deployment of base stations cannot meet their real-time changing communication needs [6]. In this case, relying only on terrestrial networks as well as satellite networks to provide maritime services has certain limitations in terms of coverage, communication capacity, and computational resources, and thus it is difficult to meet large-scale, highly reliable, and low-latency maritime communication needs.

In this context, UAV-assisted communication has gradually become an effective program. With the characteristics of flexibility and high mobility, UAVs can dynamically adjust their flight altitude and position according to real-time demand in the marine environment, thereby covering the blind areas that it is difficult to cover with traditional communication networks. Compared with satellites and terrestrial base stations, drones can be rapidly deployed to specific areas to provide high-quality localized communication coverage, especially in hot spots where communication needs are more concentrated, such as sea areas far from land and near offshore platforms [7,8]. UAVs are also able to adjust flight routes and altitudes according to real-time communication needs, reducing signal attenuation and ensuring communication quality. In addition, UAVs can increase the line-of-sight (LOS) transmission range of the airborne base station so that it can communicate more effectively with ships, buoys, and other equipment on the sea surface, further improving the communication quality and network reliability [9,10]. On this basis, there is an urgent need to develop more efficient network architectures and deployment schemes for better performance in massive maritime data transmission and task processing.

In the maritime network environment, due to the mobility of ships and the limitation of UAV communication coverage, UAVs need to adjust their flight position and altitude in real time to ensure that they can maintain a stable communication connection with ships. In this case, it is particularly important to accurately predict the ship's route. By predicting the ship's trajectory, UAVs can plan their flight paths and adjust their positions in advance, thus optimizing the deployment and scheduling of UAVs. This not only helps to improve the response speed of UAVs in complex marine environments but also ensures the continuity and stability of communication coverage [11]. Automatic Identification Systems (AISs) can monitor ship dynamics and can obtain real-time information on the transmitted positions of hundreds of thousands of ships around the world [12]. These data can be used to effectively predict the ship's trajectory and provide UAVs with positional information for some time in the future. As a result, the UAV can make real-time adjustments more flexibly, thus significantly improving the response speed and stability of maritime communication signals, and ensuring the UAV's communication coverage capability under complex sea conditions.

This paper explores the potential benefits of UAVs in enhancing maritime coverage, with a focus on coordinating UAVs with existing satellites and ground base stations through route prediction to enable dynamic deployment and improve the overall performance of maritime communication networks. The main contributions of this paper are as follows:

1. We propose a deep learning algorithm-based voyage trajectory-prediction model for predicting the voyage trajectory of the user's vessels, which enables communication nodes to sense the user position more accurately, thereby optimizing the deployment scheme and ultimately achieving dynamic and precise coverage.
2. We designed a RUDD algorithm based on route prediction. The algorithm takes into account the computational resources, communication coverage, latency of the communication nodes, the limitations of the UAVs' battery capacity, and the mobility of the user, aiming to maximize the network coverage, reduce the total latency of the system to process the communication tasks, and reduce the energy consumption of the UAV to enable dynamic deployment.
3. We design simulation experiments to demonstrate that the RUDD algorithm significantly outperforms other basic algorithms in reducing total system cost, improving

communication coverage, and reducing system delay. We also test the algorithm under different system model parameters to evaluate its performance. Simulation results show that the proposed algorithm has better stability and confirms its applicability in marine IoT scenarios.

The rest of the paper is structured as follows. Section 2 discusses related work. Section 3 describes the system model and problem formulation. Then, Section 4 presents the design of the RUDD program process. In Section 5, the experimental performance is evaluated. Section 6 presents the conclusion.

2. Related Work

Near-Earth orbit (LEO) satellites play an important role in maritime communications, and many references in the literature have conducted specific studies on the satellite-wide-area coverage problem. For example, ref. [13] analyzes the average backhaul capacity of terrestrial satellite terminals using stochastic geometry and queuing theory and proposes a multi-layer LEO satellite constellation deployment scheme that takes satellite mobility into account and supports seamless global coverage. To address the saturation problem in available LEO space and meet the ultra-low latency requirements of future 6G, ref. [14] adopts a decomposition-aggregation approach combined with an elite strategy genetic algorithm to minimize satellite size and maximize average coverage, ensuring high robustness and reliability. To address issues such as the limitations of a single satellite deployment model, insufficient channel model accuracy, and limited communication coverage, ref. [15] proposes a multi-layer satellite deployment strategy. By deploying satellites on multi-layer concentric spheres, this strategy significantly enhances both the model's pervasiveness and the coverage capability of the satellite communication system. Meanwhile, the Shadow Rice Fading (SR) model is introduced to characterize the channel between satellites and terrestrial gateway stations, which improves the accuracy of the channel model. Although some progress has been made with existing satellite-deployment strategies, these methods usually do not take into account the deployment of drones to supplement coverage. When satellites are deployed on a large scale to extend coverage, it is not only costly but also results in higher losses during signal transmission and lower transmission rates, making it difficult to meet the demand for low latency and high reliability in maritime communications.

The mobility and flexibility of UAVs enable them to provide essential communication services to users in the edge areas of the cell as well as to users with higher requirements for communication quality. Several research papers have focused on applications and studies that rely on UAVs alone for coverage. For example, ref. [16] investigates the problem of 3D path planning for UAVs with cellular network connectivity and proposes a multi-step dueling DDQN-based algorithm for coverage maximization. Ref. [17] mainly explores the system performance of UAV-assisted networks in urban environments, focusing on the radiation gain of directional antennas. By deriving the network coverage probability based on stochastic geometric theory, the signal transmission efficiency and coverage range were significantly improved. Ref. [18] proposed a distributed algorithm based on the virtual Coulomb force and Voronoi diagram with two mobility schemes and redundant UAV dormancy strategy to minimize the number of UAVs, improve communication coverage, and save energy. Ref. [19] investigates the coverage utility and energy multi-objective optimization problem in multi-UAV communication scenarios and proposes an improved multi-objective Gray Wolf optimization algorithm. The algorithm optimizes the number, position, and speed of UAVs to maximize the coverage utility and minimize energy consumption through role determination and hybrid solution initialization strategies. Ref. [20] investigates the multi-UAV base station deployment problem considering constraints such as movement speed, energy consumption, and communication coverage radius and proposes an algorithm called Dense Multi-Intelligent Body Reinforcement Learning, which aims to maximize the communication coverage of the vehicle-mounted network. For the problem of UAV swarms performing full coverage of an area, ref. [21] proposes a path-planning algorithm realized by information exchange between UAV swarms and adopts the parallel line full coverage path to provide an effective solution for UAV swarms' full coverage path planning in a simple area. To solve the problem of wireless emergency communication in maritime emergency communication networks, ref. [22] proposed a wireless emergency communication relay system based on a tethered UAV platform, which accounts for significant advantages in rapid and flexible deployment and long-time spatial coverage. Ref. [23] investigated the deployment of UAVs to optimize coverage quality after a disaster or during episodic events. A decentralized deployment algorithm based on weighted Voronoi cells is proposed, aiming to minimize the average distance between UAVs and users while maintaining the connectivity between UAVs and fixed base stations to improve coverage performance.

Although UAVs are playing an increasingly important role in communication coverage, a single UAV coverage solution may not be able to provide stable and long-lasting services in some scenarios due to energy constraints, coverage limitations, and other issues. For this reason, combining UAVs with communications from satellites and traditional base stations to build a synergistic and complementary multilevel communication network will be able to address the limitations in existing research. Much of the literature examines how UAVs can be deployed into existing satellite–ground integrated networks to improve maritime network coverage. For example, considering the impact of the distance between oceanic surface stations and the coastline on the coverage performance, ref. [24] investigates a wide-range maritime communication architecture based on SAGSIN, which analyzes the random distribution of oceanic surface stations on the ocean surface and improves the coverage probability of oceanic surface stations. Ref. [25] proposed a hybrid satellite–UAV–terrestrial network based on NOMA technology with a joint power allocation scheme to maximize the rate and coverage of the offshore network. Ref. [26] combines the advantages of wide coverage of satellites with the high capacity of shore-based systems and deploys UAVs to enhance the coverage of a hybrid satellite–terrestrial offshore communication network, while jointly optimizing the UAV trajectory and flight transmit power. Ref. [27] describes a hybrid satellite–UAV–terrestrial network for maritime communications that achieves extensive coverage and energy efficiency at sea by coordinating different communication links. While these studies consider factors such as energy consumption, communication latency, and coverage, they mostly ignore the computational resource constraints of the UAVs themselves and the dynamic mobility of maritime users.

Considering the impact of maritime user mobility on the deployment of communication nodes, prediction algorithms have been introduced to forecast ship trajectories, enabling more effective dynamic deployment. Many studies have focused on trajectory-prediction algorithms for ships. For example, ref. [28] investigated how to predict ship trajectories in the inner harbor of Busan port using AIS data and deep learning techniques, solving the problem of irregular intervals of AIS data through linear interpolation, and improving the accuracy of route prediction in a complex port environment. Ref. [29] proposed a bi-directional data-driven trajectory-prediction method based on AIS spatio-temporal data, constructed an encoder–decoder network driven by forward and reverse integrated historical trajectories, and predicted ship trajectories by fusing the characteristics of the sub-networks. Ref. [30] proposed a multi-gated attention encoder–decoder network that significantly improves the accuracy of ship trajectory prediction. The scheme combines the LSTM network with Gated Recurrent Units and an attention mechanism and enhances the generalization ability and robustness of the model by introducing a soft-threshold residual structure to handle sparse features. Ref. [31] proposes a deep learning-based framework for ship trajectory prediction, consisting of two models: Differential Long Short-Term Memory (DLSTM) and Enhanced DLSTM with Reference Trajectory Correction (Ref-DLSTM). These models are used for cases without and with reference trajectories, respectively, effectively reducing prediction errors. Ref. [32] proposes a ship trajectory-prediction algorithm called Deep Bidirectional Information Empowerment, which utilizes an integrated network and attention mechanism. It combines the strengths of bidirectional long short-term memory and bidirectional gated recurrent unit networks, optimizing the weights of both network units through the attention mechanism to enhance prediction accuracy and efficiency. Considering the multi-density distribution characteristics of trajectory data, a multi-density adaptive trajectory-clustering algorithm is proposed in ref. [33], which determines the input parameters adaptively and introduces a trajectory direction identification mechanism to make it perform better in dealing with complex trajectory-clustering problems. To address the heterogeneity of vessel motion patterns, ref. [34] proposes a generalized vessel trajectory-prediction method based on clustering. This method uses historical AIS data to cluster route patterns for each vessel type, considering spatial and heading attributes as well as environmental factors, thereby improving the accuracy and computational efficiency of trajectory prediction. To overcome the reliance on historical position data while ignoring key factors like speed and heading in voyage trajectory prediction, ref. [35] proposes a novel ship trajectory-prediction model based on a sequence-to-sequence structure. This model integrally considers multifaceted ship information and improves prediction accuracy through a multi-semantic encoder and a type-oriented decoder.

User ships are mobile when operating at sea, and route prediction for mobile users can effectively anticipate their future locations. This aids in the efficient deployment of communication nodes and enables dynamic coverage for less computationally demanding or non-urgent tasks. The lack of existing research in this area will affect the effectiveness of dynamic coverage to a certain extent. Therefore, the research in this paper focuses on studying a model that can realize the prediction of ship trajectories, and at the same time can improve the coverage of communication as well as the UAV flight trajectory, reduce the communication delay, and improve the quality of service for users.

3. System Modeling

3.1. Network Model

We consider a network architecture consisting of mobile users, UAVs, terrestrial base stations (TBSs), and satellites. TBSs are deployed in coastal areas to provide communication services to users in coastal waters. The broadband coverage of TBSs is usually limited due to high non-line-of-sight path loss. Outside the TBS coverage area, satellites deployed in space orbit provide communication links. Ships equipped with expensive high-gain antennas can be guaranteed broadband service. However, for low-end ships without high-gain antennas, it is still difficult to enjoy broadband service even within the satellite coverage area. To fill this gap, we use drones to provide broadband services. More specifically, if a mobile user requires a high-rate communication service (e.g., videoconferencing) from t_s to t_e, the communication request will be sent from the mobile user to the nearest TBS and then transmitted to the central processor. The central processor selects an idle UAV and prepares the idle UAV to serve the mobile user. The UAV will fly along the optimized trajectory to serve the user from t_s to t_e. After completing the high-speed communication service, the mobile user will associate with the nearest TBS at the moment t_e, and the UAV will return to the coast. Consider a target coverage area with network services from an LEO satellite, a terrestrial base station, and m drones, and there are n user ships in the area, which are serviced with $m \in \{1, 2, 3, \cdots, M\}$, and $n \in \{1, 2, 3, \cdots, N\}$, as shown in Figure 1. The satellite serves as the main base station, which mainly handles computing and communication tasks for the more concentrated users at sea. Each UAV is equipped with a computational processor that can handle simple computational tasks. The transmission time is time-slotted, and it is assumed that one information-transmission cycle is divided into T time slots, each denoted by t, where $t \in \{1, 2, 3, \cdots, T\}$. The user moves at a speed of u_{speed_n}, which satisfies $0 \leq u_{speed_n} \leq V_{max}$, where V_{max} denotes the maximum speed of vessel movement.

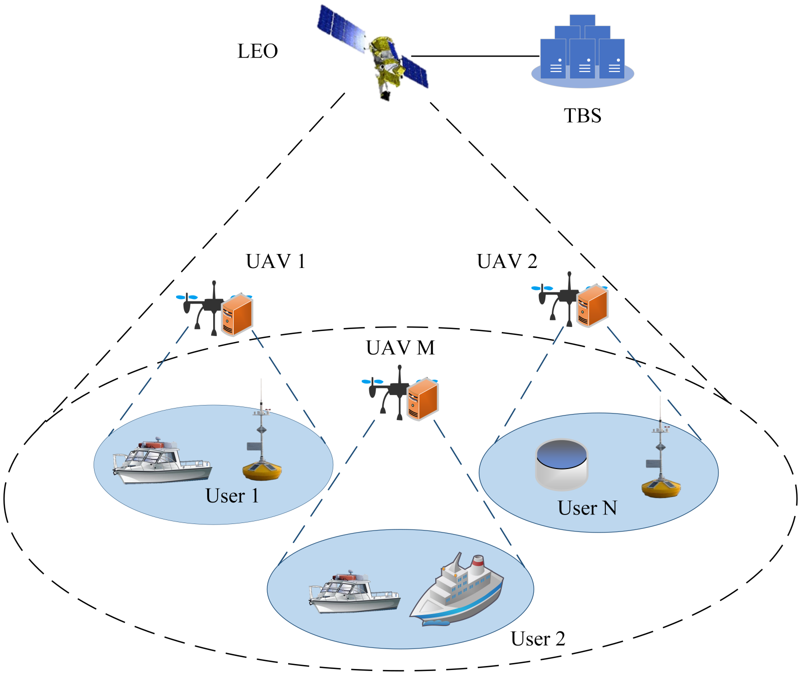

Figure 1. Maritime communication network model.

3.2. Network Communication Model

At the t-th time slot, the position of the m-th UAV can be expressed as:

$$w_m(t) = (x_m(t), y_m(t), h) \tag{1}$$

$$x_m(t+1) = x_m(t) + \cos\theta\cos\phi d_m(t) \tag{2}$$

$$y_m(t+1) = y_m(t) + \sin\theta\cos\phi d_m(t) \tag{3}$$

where $x_m(t)$ denotes as the horizontal coordinate of the m-th UAV at t time slots and $y_m(t)$ denotes the vertical coordinate of the m-th UAV at the time of t time slots, and h denotes the height of the m-th UAV from the sea surface at the time of t time slots. In this paper, the altitude of the flight varies between h_1 and h_2. $x_m(t+1)$ denotes the horizontal coordinate of the m-th UAV at $t+1$ time slots, and $y_m(t+1)$ denotes the vertical coordinate of the m-th UAV at $t+1$ time slots. $d_m \in [d_{\min}, d_{\max}]$ denotes the distance flown by the drone at a one-time slot. $d_{\max}$ denotes the maximum flight distance of the drone in a one-time slot. $\theta \in [0, 2\pi]$ denotes the horizontal flight direction.

The linear distance between UAV m and user n is as follows, where $x_n(t)$ denotes the horizontal coordinate of user n at t time slots, and $y_m(t)$ denotes the vertical coordinate of user n at t time slots:

$$L_{(m,n)}(t) = \sqrt{h_m(t)^2 + (x_m(t) - x_n(t))^2 + (y_m(t) - y_n(t))^2} \tag{4}$$

In our model, it is assumed that the satellite and terrestrial base station TBS are relatively stationary within the shorter time slot T_{set}. At the moment t, the position of the satellite is denoted as $(x_{\text{Leo}}(t), y_{\text{Leo}}(t), h_{\text{Leo}}(t))$, and the position of the ground base station is denoted as $(x_{\text{tbs}}(t), y_{\text{tbs}}(t))$. The straight-line distance between the satellite, ground base station, and user n is denoted as:

$$L_{\text{Leo},n}(t) = \sqrt{h_{\text{Leo}}(t)^2 + (x_{\text{Leo}}(t) - x_n(t))^2 + (y_{\text{Leo}}(t) - y_n(t))^2} \tag{5}$$

$$L_{\text{tbs},n}(t) = \sqrt{(x_{\text{tbs}}(t) - x_n(t))^2 + (y_{\text{tbs}}(t) - y_n(t))^2} \tag{6}$$

Assuming that there is no mission migration between UAVs, satellites, and base stations, and that the minimum channel capacity C and bandwidth B of each device are known, we can find the minimum signal-to-interference-plus-noise-ratio (SNR) according to Shannon's formula:

$$\zeta_{\mathrm{UAV,min}}(t) = 2^{\frac{C_{\mathrm{UAV,min}}(t)}{B_{\mathrm{UAV}}(t)}} - 1 \tag{7}$$

$$\zeta_{\mathrm{leo,min}}(t) = 2^{\frac{C_{\mathrm{leo,min}}(t)}{B_{\mathrm{leo}}(t)}} - 1 \tag{8}$$

$$\zeta_{\mathrm{tbs,min}}(t) = 2^{\frac{C_{\mathrm{tbs,min}}(t)}{B_{\mathrm{tbs}}(t)}} - 1 \tag{9}$$

Assume that all UAVs share the same frequency band, so the UAVs interfere with each other in downlink transmissions [36]. Meanwhile, it is assumed that the transmit power P_t from all UAVs to the user and the channel gain $G_0 = \frac{(G_t G_r)}{r^{\alpha}_{\mathrm{UAV,max}}}$ are the same, so the signal-to-noise ratio ς can also be expressed as the following equation, where n^2 denotes the noise power at the receiving end:

$$\zeta = \frac{P_{\mathrm{UAV},t} G_0}{(M-1) P_{\mathrm{UAV},t} G_0 + n^2} \tag{10}$$

Therefore, the maximum coverage radius of the UAV can be derived as:

$$r_{\mathrm{UAV,max}}(t) = \left[\frac{P_{\mathrm{UAV},t} G_{\mathrm{UAV},t} G_{\mathrm{UAV},r} [1 - (M-1)\zeta_{\mathrm{UAV,min}}(t)]}{\zeta_{\mathrm{UAV,min}}(t) n^2} \right]^{\frac{1}{\alpha}} \tag{11}$$

Since this system assumes that there is only one satellite and base station and they are in separate frequency bands, the interactions between the UAV, satellite, and base station are negligible when calculating the signal-to-noise ratio. Therefore, we can find out the SNR $\varsigma = \frac{P_t G_t G_r}{r^{\alpha}_{\mathrm{max}} n^2}$, and the maximum coverage radius of the satellite and the base station are, respectively:

$$r_{\mathrm{leo,max}}(t) = \left(\frac{P_{\mathrm{leo},t} G_{\mathrm{leo},t} G_{\mathrm{leo},r}}{\zeta_{\mathrm{leo,min}}(t) n^2} \right)^{\frac{1}{\alpha}} \tag{12}$$

$$r_{\mathrm{tbs,max}}(t) = \left(\frac{P_{\mathrm{tbs},t} G_{\mathrm{tbs},t} G_{\mathrm{tbs},r}}{\zeta_{\mathrm{tbs,min}}(t) n^2} \right)^{\frac{1}{\alpha}} \tag{13}$$

In each time slot, if the distance $L_{(m,n)}(t)$ between user n and UAV m is less than or equal to the communication coverage radius $r_{\mathrm{UAV,max}}$ of UAV, a communication link can be established. Similarly, a communication link can be established if the distance $L_{\mathrm{Leo},n}(t)$ between user n and the satellite is less than or equal to the communication coverage radius $r_{\mathrm{leo,max}}$ of the satellite. A communication link can be established if the distance $L_{\mathrm{tbs},n}(t)$ between user n and the TBS is less than or equal to the communication coverage radius $r_{\mathrm{tbs,max}}$ of the terrestrial base station.

3.3. Time-Delay Model

Due to the limited computational resources of each device, when the computational resources of satellites and base stations are insufficient to meet the computational needs of users, UAVs are required to perform auxiliary computations. However, a non-negligible delay is generated in the process of providing communication and computation services to users, so this section focuses on the delay model in the maritime communication scenario.

Assuming that at the t-th time slot, if the maritime user n is within the communication coverage of both the satellite and the base station, and both of them have sufficient computational resources, the user equipment will prioritize establishing a communication connection with the satellite, if the satellite resources are insufficient, it will choose the base station to communicate; if the base station also fails to satisfy the demand, the equipment will try to communicate with the nearest UAV. When the mission ends, the device will recover the amount of computation required for that user's mission. In marine environments, the flight delay and transmission delay are affected by environmental factors such as wind speed, humidity, and airflow, which are assumed to be represented by a discount factor $\omega \in (0,1)$. Additionally, the complexity of the computational task is represented by a factor k $(k > 0)$.

When the amount of computation required a_n for a user task is less than the computational power $a_{\text{leo,ava}}$ available to the satellite, the user device will establish a communication connection with the satellite. Assume that the channel transmission rate between the satellite and user n is $r_{\text{leo,n}}(t)$. The amount of data to be transmitted by user device n is $q_{\text{Leo,n}}$, and the amount of task data that can be processed by the satellite per second is f_{Leo}. Then the total delay of the satellite can be expressed as Equation (14):

$$d_{\text{Leo,n}}(t) = d_{\text{trans leo,n}}(t) + d_{\text{comp leo,n}}(t), \tag{14}$$

where the task transmission time $d_{\text{trans leo,n}}(t)$ and computational delay $d_{\text{comp leo,n}}(t)$ can be expressed as Equations (15) and (16), respectively:

$$d_{\text{trans leo,n}}(t) = \frac{q_{\text{Leo,n}}}{\omega \cdot r_{\text{Leo,n}}(t)}, \tag{15}$$

$$d_{\text{comp leo,n}}(t) = \frac{k \cdot q_{\text{Leo,n}}}{f_{\text{Leo}}}, \tag{16}$$

Similarly, when the amount of computation required a_n for a user task is less than the computational power $a_{\text{tbs,ava}}$ available at the base station, the user device will establish a communication connection with the base station. The total delay between the base station and the associated user n can be expressed as Equation (17), where the task-transfer time $d_{\text{trans tbs,n}}(t)$ and computational delay $d_{\text{comp tbs,n}}(t)$ are given by Equations (18) and (19), respectively:

$$d_{\text{tbs,n}}(t) = d_{\text{trans tbs,n}}(t) + d_{\text{comp tbs,n}}(t), \tag{17}$$

$$d_{\text{trans tbs,n}}(t) = \frac{q_{\text{tbs,n}}}{\omega \cdot r_{\text{tbs,n}}(t)}, \tag{18}$$

$$d_{\text{comp tbs,n}}(t) = \frac{k \cdot q_{\text{tbs,n}}}{f_{\text{tbs}}}, \tag{19}$$

When the amount of computation required a_n for a user task is less than the computational power $a_{\text{m,ava}}$ available to UAV m, the user device will establish a communication connection with UAV m. For the flight delay $d_{\text{fly m,n}}(t)$ between UAV m and user device n, which is affected by the maritime environment, it can be expressed as Equation (20):

$$d_{\text{fly m,n}}(t) = \frac{L_{m,n}(t)}{\omega \cdot d_m}, \tag{20}$$

Therefore, the total delay between UAV m and the associated user n can be expressed as Equation (21):

$$d_{m,n}(t) = d_{\text{trans m,n}}(t) + d_{\text{fly m,n}}(t) + d_{\text{comp m,n}}(t), \quad n \in \mu, \tag{21}$$

where the mission transfer time $d_{\text{trans m,n}}(t)$ and computational delay $d_{\text{comp m,n}}(t)$ can be expressed as Equations (22) and (23), respectively:

$$d_{\text{trans m,n}}(t) = \frac{q_{m,n}}{\omega \cdot r_{m,n}(t)}, \tag{22}$$

$$d_{\text{comp m,n}}(t) = \frac{k \cdot q_{m,n}}{f_m}, \tag{23}$$

Finally, we define the total system delay $d_{\text{total}}(t)$ as Equation (24). Assuming that the quality of service (QoS) delay metric for user n is α, it needs to satisfy the threshold α to meet the demand of user n. Otherwise, the current device cannot satisfy the demand of user n:

$$d_{\text{total}}(t) = \begin{cases} d_{\text{Leo,n}}(t), & \text{if connected to satellite} \\ d_{\text{tbs,n}}(t), & \text{if connected to base station} \\ d_{m,n}(t), & \text{if connected to UAV} \end{cases} \tag{24}$$

3.4. Energy-Consumption Model

Since UAVs are limited by battery capacity, in the case of high energy consumption, UAVs may find it difficult to fulfill their tasks. Therefore, this subsection mainly describes the energy-consumption model of UAVs. The total energy consumption $E_{m,n}(t)$ of UAV m providing services to user n can be expressed as the sum of the flight energy consumption and the hover energy consumption, as shown in Equation (25):

$$E_{m,n}(t) = E_{\text{fly m,n}}(t) + E_{\text{hover m,n}}(t), \tag{25}$$

Considering that the UAV is in a hovering state when providing computational resources for user requests, the hovering energy consumption $E_{\text{hover m,n}}(t)$ of the UAV can be calculated as follows, where $P_{\text{hover m}}(t)$ is the power consumption of the UAV when hovering, m_{uav} is the mass of the UAV, g is the gravitational acceleration, ρ_{env} is the air density in the marine environment, and A is the total swept area of the rotor blades of the UAV:

$$P_{\text{hover}}(t) = \sqrt{\frac{(m_{\text{uav}} \cdot g)^3}{2 \cdot \rho_{\text{env}} \cdot A}}, \tag{26}$$

$$E_{\text{hover m,n}}(t) = P_{\text{hover}}(t) \cdot \left(d_{\text{trans m,n}}(t) + d_{\text{comp m,n}}(t)\right), \tag{27}$$

According to the user position obtained from the user trajectory-prediction model in Section 4.1, after solving for the optimal position of the UAV, the flight energy consumption of the UAV moving to that position can be calculated by the following equation. Here, $P_{\text{fly}}(t)$ is the UAV flight power consumption, and β is the air resistance experienced by the UAV in the marine environment:

$$P_{\text{fly}}(t) = P_{\text{uav hover}}(t) + \beta \cdot d_m^3, \tag{28}$$

$$E_{\text{fly}}(t) = P_{\text{fly}}(t) \cdot d_{\text{fly m,n}}(t), \tag{29}$$

3.5. Formulation of the Problem

A user can establish a connection with a device when the user is located within the communication range of the respective device, and the required computation does not exceed the available computational capacity of the device, while satisfying the delay and energy-consumption constraints (i.e., the maximum delay limit $d_{\max}$ and the UAV battery capacity limit $E_{\max}$). The establishment of a connection between a device and a user can be represented as follows:

$$X_n = \begin{cases} 1, & \text{if } L \le r_{\max} \text{ and } a_n \le a_{\text{ava}} \text{ and } d_{\text{total}}(t) \le d_{\max} \text{ and } E_{m,n}(t) \le E_{\max} \\ 0, & \text{else} \end{cases} \tag{30}$$

To evaluate the coverage of the overall network deployment, a parameter Q is defined to quantify the coverage performance of the entire system as follows:

$$Q = \frac{1}{N} \sum_{n=1}^{N} X_n \tag{31}$$

In order to optimize mobile communication network coverage and ensure efficient collaboration between satellites and terrestrial base stations for comprehensive communication support for the maritime user community, this study designs a dynamic deployment strategy for UAVs. This strategy aims to provide reliable and immediate communication support for users in emergency situations or for those who cannot be covered by conventional networks. Based on a comprehensive consideration of the computational resources of each communication node, the coverage connectivity and communication delay for mobile users, and the battery capacity of the UAV, the objective is to minimize the total system cost and maximize user communication coverage. The optimization problem is modeled as follows:

$$\begin{aligned} &P1 \ \max \sum_{n=1}^{N} X_n - \lambda \left[\sum_{m=1}^{M} \Big(E_{\text{fly},m,n}(t) + E_{\text{hover},m,n}(t) \Big) + \sum_{n=1}^{N} d_{\text{total}}(t) \right] \\ \text{s.t.} \quad & C1 : L \leq r_{\max}, \\ & C2 : d_{\text{total}}(t) \leq d_{\max}, \\ & C3 : E_{\text{fly m,n}}(t) + E_{\text{hover m,n}}(t) \leq E_{\max}, \\ & C4 : a_n \leq a_{\text{ava}}, \\ & C5 : X_n \in \{0,1\}, \forall n \end{aligned} \tag{32}$$

4. Proposed Algorithm

In this section, we formulate the dynamic UAV-deployment problem based on route prediction. The goal is to predict the sailing trajectory of a user ship by a reinforcement learning algorithm and dynamically deploy UAVs to maximize user coverage based on the sailing trajectory under constrained energy consumption, latency, and computation. We propose a dynamic UAV-deployment scheme for changing user dynamics that combines deep reinforcement learning algorithms for route prediction as well as for UAV deployment.

4.1. Ship Trajectory-Prediction Algorithm Based on Improved LSTM

To improve the accuracy of ship route prediction, this section proposes an LSTM model optimized based on the Sparrow Search Algorithm (SSA) for optimal prediction of AIS datasets. The LSTM performs well in dealing with time-series prediction tasks (e.g., ship route prediction), whereas by mimicking the behavior of sparrow foraging, the SSA can be optimized in a complex, high-dimensional search space hyperparameters of LSTM. Therefore, combining SSA with LSTM can automatically optimize the hyperparameters to improve the prediction performance of the model.

The AIS dataset contains data such as latitude, longitude, heading, and speed of different numbered vessels at each moment in the past period, so we performed preprocessing operations on the AIS dataset. First, we performed feature selection and extracted the four features of latitude, longitude, speed (Sog), and heading (Cog) from the ship trajectory data as the input feature matrix x, and the target value y was set to be the latitude and longitude of the next time step. Then, the input data $X_t = [X_{t-N}, X_{t-N+1}, \ldots, X_{t-1}]$ and output data $Y_t = [lat_t, lon_t]$ are constructed into a time series, and the data of the past N time steps are used to predict the target value of the next time step, to capture the temporal features and dynamics. To avoid the impact of magnitude differences between feature values on model training, we use min-max normalization to scale the input features and target values into the interval [0, 1].

When inputting the input features into the input layer of the LSTM, we converted them into a 3D tensor that can represent the feature data for multiple time steps, i.e., $[\text{number of samples}, \text{time step } N, \text{number of features}]$, where the number of samples represents the number of time series samples that can be created for the entire dataset, and the number of features is represented as four features (latitude, longitude, speed, and heading) for each time point. In the hidden layer, the LSTM unit processes the input data through a memory and forgetting mechanism to capture the long-term dependencies in the time series. Considering the model performance, training efficiency, and avoiding the risk of overfitting, two LSTM hidden layers are used in the model, and the number of neurons in the first and second layers is set to h_1 and h_2, respectively, which will be optimized by the SSA. The LSTM updates the hidden state through the gating structure, and the state of the hidden layer's output at the last time step, h_t, is denoted by Equation (33), where W_{hid} is the weight matrix of the hidden layer, b_{hid} is the bias term, and f is the nonlinear activation function of the LSTM. In the output layer, the model maps h_t to the predicted longitude and latitude, which is calculated as in Equation (34), where W_{out} is the weight matrix of the output layer and b_{out} is the bias term.

$$h_t = f(W_{\text{hid}} \cdot X_t + b_{\text{hid}}) \tag{33}$$

$$Y_t = W_{\text{out}} \cdot h_t + b_{\text{out}} \tag{34}$$

Finally, we use the mean square error to measure the difference between the predicted and true values of the model, continuously minimize the loss function, and update the weights and biases through the Adam optimizer to improve the prediction accuracy. To prevent overfitting, we introduce L2 regularization in the training process of LSTM networks. The complexity of the model is limited by adding a penalty term λ proportional to the sum of squares of the weights to the loss function, which improves the generalization performance of the model and reduces the risk of overfitting on the training data, and the formula is specified as follows:

$$L_{\text{total}} = L_{\text{loss}} + \lambda(\|W_{\text{hid}}\|^2 + \|W_{\text{out}}\|^2) \tag{35}$$

Here, we optimize the hyperparameters of the LSTM, i.e., the learning rate, epoch, and the number of hidden layer neurons. First, the SSA algorithm requires the initialization of a population, where each individual represents a set of hyperparameter combinations to be optimized, and the locations of these individuals are randomly generated within the upper and lower ranges $[lb, ub]$. Each individual's performance is evaluated by a fitness function (i.e., mean square error), where a smaller fitness value indicates that the hyperparameter combination is more effective. We categorize the updating strategies of individual positions into three categories; in each iteration, the top 50% of individuals update their positions using strategy 1, i.e., the individual positions are close to the current optimal individual, with the updating Equation (36), where $x_{\text{best}}(t)$ is the position of the current optimal individual, and ϵ is a uniformly distributed random number within the interval of $[-1, 1]$, which is used to increase search diversity and avoid falling into local optimality. For the last 50% of individuals, update strategy 2 is used to generate new positions through a global random search, and the update formula is Equation (37), where L_j and U_j are the upper and lower limits of the j-th dimension, and α_j is a random number in the interval $[0, 1]$. After each position update, the algorithm performs boundary processing to ensure that the position $x_{(i,j)}$ of each dimension stays within the upper and lower limits $[L_j, U_j]$, which is handled by Equation (38).

$$x_i(t+1) = x_{\text{best}}(t) + \epsilon \cdot |x_i(t) - x_{\text{best}}(t)| \tag{36}$$

$$x_i(t+1) = L_j + \alpha_j \cdot (U_j - L_j) \tag{37}$$

$$x_{(i,j)}(t+1) = \begin{cases} L_j, & \text{if } x_{(i,j)}(t+1) < L_j \\ U_j, & \text{if } x_{(i,j)}(t+1) > U_j \\ x_{(i,j)}(t+1), & \text{otherwise} \end{cases} \tag{38}$$

After each policy update, the algorithm will recalculate the fitness value of each individual and record the fitness of the current optimal individual. At the end of several iterations, the algorithm outputs the individual with the smallest fitness value, i.e., the optimal hyperparameter combination. The specific pseudo-code of the algorithm is as Algorithm 1:

Algorithm 1 Optimization of the LSTM model for ship route prediction using the SSA algorithm

```
Input ship data (latitude, longitude, SOG, COG)
Initialize the LSTM model hyperparameter spaces $lb$ and $ub$;
Initialize population size, number of iterations, individual position population;
The fitness MSE was calculated for each individual in the population;
for each episode do
    Find the current optimal individual $x_{best}$;
    For the top 50% of individuals, the individual's position was adjusted using the Equation (36) update strategy;
    For the latter 50% of individuals, the location was updated using a global random search via the Equation (37);
    Performs boundary processing by the Equation (38) in the updated position;
    Reassess the fitness of each individual in the population;
end for
Output the optimal individual, i.e., the optimal LSTM hyperparameter combination;
Constructing the model: Define an LSTM model with 2 hidden layers and compile the model using the Adam optimizer and MSE loss function.
Training LSTM models using optimal hyperparameter combinations;
Predicting the next trajectory of a vessel using a trained LSTM model;
Evaluate the prediction accuracy and calculate the mean square error;
```

4.2. UAV Dynamic Deployment Algorithm

Traditional UAV-deployment methods only consider the limitations of the UAV's energy consumption and latency, without taking into account the mobility of the user nodes and the UAV's computational resources. Therefore, in this section, a deep reinforcement learning algorithm is used to deploy drones dynamically. Trajectory prediction is used to understand the user distribution in advance so that decision-makers can make better decisions to maximize the coverage with less delay and energy consumption. At each time slot, the agent collects ship prediction data, dynamically assigns drones, and develops deployment strategies based on predicted device locations as well as task volumes.

We convert the UAV-deployment problem into a Markov Decision Process (MDP) problem, defined by the tuple (S, A, p, r), where the state space S and action space A are continuous. Specifically, the Markov chain is denoted by $(s_t, a_t, s_{(t+1)}, r_t)$, where each parameter consists of an output state $s_t \in S$ at moment t, an action $a_t \in A$, an output state $s_{(t+1)} \in S$ at moment $t+1$, and a reward $r_t \in r$. The output state s_t is received from the environment to the intelligent agent at moment t. The intelligent agent uses an internal policy function to compute the probabilities of each action and selects an action a_t based on these probabilities. Applying action a_t to the environment yields state $s_{(t+1)}$. Finally, the environment will combine the action a_t, the state transitions, and the reward function to compute the reward value r_t under action a_t and return it to the intelligent agent so that the agent can optimize its future action strategy based on the reward situation. The complete trajectory of the interaction process with the environment can be represented by Equation (39):

$$\tau = \{s_0, a_0, s_1, a_1, \ldots, s_t, a_t\} \tag{39}$$

The state space S consists of $(3M + 6)$ states, where M represents the number of available drones. Based on the trajectory-prediction algorithm described in Section 4.1, we can predict the movement trajectories of uncovered users over some time. At the moment t, by performing K-means clustering analysis on the user coordinates, we can determine the location of the user's aggregation center point at that moment, whose 3D coordinates correspond to three states in the state space. In addition, the other three states describe the decision effects at moment t, including the total number of covered users, the accumulated delay, and the overall energy consumption, respectively. Meanwhile, $3M$ represents the remaining battery capacity, available computing resources, and the distance of the UAVs relative to the center point of the user distribution at moment t, respectively, for M UAVs. The action space A consists of $3M$ consecutive action variables, each of which represents the amount of adjustment of the coordinates of the M UAVs in the 3D space at the moment t. Together, these action variables determine the change of the UAV's position at the next moment, thus enabling efficient optimization of user coverage.

The reward function is designed as:

$$r_{\text{total}}(t) = \sigma \sum_{n=1}^{N} X_n + \varsigma \frac{\sum_{n=1}^{N} X_n}{\sum_{m=1}^{M} \left(E_{\text{fly m,n}}(t) + E_{\text{hover m,n}}(t) \right)} + \tau \frac{\sum_{n=1}^{N} X_n}{d_{\text{total}}(t)} \tag{40}$$

where σ, ς, and τ are the moderating factors used to regulate the reward weights of each component. As the number of covered users increases, the inverse of the average energy consumption and the inverse of the average delay decrease with the increase in users. To achieve the reduction of delay and energy consumption while maximizing the number of covered users, the larger this reward function, the better the performance of the algorithm.

The traditional strategy gradient approach leads to unstable training due to the excessive magnitude of strategy updates, while trust domain strategy optimization (TRPO) has a high computational complexity despite the introduction of constraints. PPO simplifies the implementation of strategy updates by introducing the clip mechanism, and at the same time effectively restricts the magnitude of the strategy updates to avoid strategy collapses during the training process. The PPO algorithm mainly employs the strategy gradient approach to avoid strategy collapses during the training process by maximizing the expectation of the dominance function to optimize the strategies, and to guide the intelligent agent to take actions that can improve the coverage and reduce the delay. In particular, it calculates the ratio of the probability of the old and new strategies and introduces the CLIP function to ensure that the new strategy improves within the scope of not deviating from the old strategy while limiting the update magnitude. The strategy objective function is computed as in Equation (41), and the ratio of the old and new strategy probabilities r_t is as in Equation (42), where $\pi_\theta(a_t|s_t)$ is the output probability of the strategy under the parameter θ, and $\pi_{\theta_{\text{old}}}(a_t|s_t)$ is the output probability of the old strategy.

$$L(\theta) = \mathbb{E}_t\left[\min\left(r_t(t) A^{\pi_{\theta_{\text{old}}}}(s_t, a_t), \text{clip}(r_t(t), 1-\epsilon, 1+\epsilon) A^{\pi_{\theta_{\text{old}}}}(s_t, a_t)\right)\right] \tag{41}$$

$$r_t(t) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{42}$$

The input of the value network is the current state s_t and the output is the value of the current state. The loss function of the value is mainly used to train the value network to minimize the difference between the value function and the actual return. The closer its predicted state value is to the actual return advantage estimate $A^{\pi_{\theta_{\text{old}}}}$, the more accurate it is, where the value loss function is as in Equation (43), and the dominance function $A^{\pi_{\theta_{\text{old}}}}$ is as in Equation (44), where $V_{\theta_v}(s_t)$ is the output of the value network, $\hat{V}_t(s_t)$ is the target value, and $r(t)$ denotes the instantaneous reward at the moment of time t. To encourage strategies to explore different actions and to prevent them from converging to a deterministic strategy too early and thus losing the ability to explore the environment, we employ entropy regularization, which is shown in Equation (45).

$$L_{\text{value}}(\theta_v) = \mathbb{E}_t\left[\left(V_{\theta_v}(s_t) - \hat{V}_t(s_t)\right)^2\right] \tag{43}$$

$$A^{\pi_{\theta_{\text{old}}}} = \sum_{t'=t}^{T} (\gamma\lambda)^{t'-t} \left(r(t) + \gamma V_{\theta_v}(s_{(t+1)}) - V_{\theta_v}(s_t) \right) \tag{44}$$

$$S(\pi_\theta) = -\sum_a \pi_\theta(a_t|s_t) \log \pi_\theta(a_t|s_t) \tag{45}$$

The total loss function of the whole algorithm is a weighted summation of three parts, namely, the strategy objective function, value loss, and entropy regularization term. By appropriately adjusting the weights of these three parts, the total loss function can guide the algorithm to find the optimal balance between convergence, accuracy, and the exploratory nature of the strategy, and its formula is shown in Equation (46).

$$L_{\text{total}}(\theta) = L(\theta) - c_1 L_{\text{value}}(\theta_v) + c_2 S(\pi_\theta) \tag{46}$$

The specific pseudo-code of the algorithm is as Algorithm 2:

Algorithm 2 Reinforcement learning model for maritime communication service coverage using RUDD algorithm

```
Initialize the strategy network $\pi$ and the value network $V$;
Initialize the experience cache pool $D$;
Initialize the optimized LSTM model to predict vessel trajectories;
Initialize the communication service coverage optimization algorithm PPO;
for each episode do
    Initialize the environment state $s$, set the initial state to contain information about the satellite, the UAV, and the ground base station, as well as information about the relative position of the UAV and the target vessel;
    for each step do
        Adopting the current policy $\pi$, action $a$ is selected according to state $s$;
        Performing action $a$, the environment returns a new state $s'$ and reward $r$;
        Store the experience ($s$, $a$, $r$, $s'$, relative positions of ships and UAVs, number of covered ships, time delay) in $D$;
        Update status $s \leftarrow s'$;
    end for
    Update the policy network $\pi$ and the value network $V$ using the data in the cache pool $D$;
    Sample a batch of data from $D$;
    Calculate the advantage function $A^{\pi_{\theta_{\text{old}}}}$ for each action;
    Updating the strategy network to maximize expected reward and entropy and reduce action bias;
    Update the value network to minimize the value function error;
    Empty cache pool $D$;
    Predicting the next position of the target vessel using the LSTM model
    The predicted locations are fed into the PPO algorithm to optimize the communication service coverage;
    Update the policy network $\pi$ to consider the optimization results of the PPO algorithm;
end for
```

5. Simulation Results and Discussion

In this section, we show the results of evaluating the performance of the proposed algorithm through simulation experiments. We provide a detailed evaluation and analysis of the convergence of the algorithm under different hyperparameter settings. We also compare the performance of the proposed algorithm with other Deep Reinforcement Learning (DRL) algorithms in terms of system cost, communication coverage, and delay performance. The experimental results show that our algorithm enables more accurate route prediction and can achieve about 95% user coverage in marine environments while ensuring that the average latency of user response is around 1 s. The algorithm is implemented in the Visual Studio Code (VSCode) environment, using the Python programming language and integrating TensorFlow and PyTorch, two popular deep learning frameworks, to improve the execution efficiency and performance of the algorithm.

5.1. Assessment of Route-Prediction Performance

Figure 2 shows the performance of different algorithms in route prediction. From the figure, we can see the difference between the actual trajectory and the predicted trajectory of each model, as well as the prediction accuracy of each algorithm. Figure 2a compares the performance of the Multilayer Perceptron algorithm (MLP), LSTM, and SSA-LSTM in actual trajectory prediction. The MLP model has a larger prediction error, whereas the LSTM is the next closest, and the SSA-LSTM has the closest prediction result to the actual trajectory, which demonstrates significant accuracy and stability. Since MLP cannot process time-series data, it is unable to capture the time dependence and dynamic changes in the trajectories. As a result, the deviation between the predicted and actual trajectories of MLP is large, as shown in Figure 2b. With the change of trajectory, the error of MLP gradually increases, which has some limitations in complex trajectory prediction. Compared with MLP, LSTM can capture the long and short-term dependencies in the time series through the memory unit and gating mechanism, which is adaptive to the trajectory data, but still has deviations in the rapidly changing regions. As shown in Figure 2c, the performance of LSTM without hyper-parameter optimization is improved but not stable enough. SSA optimizes the hyper-parameters of LSTM to work in the best configuration, which significantly improves the prediction accuracy. As shown in Figure 2d, the prediction results of SSA-LSTM almost completely overlap with the actual trajectories with small errors, showing excellent performance. Overall, the SSA-optimized LSTM model outperforms the MLP and the unoptimized LSTM in terms of prediction accuracy and stability and can better adapt to complex trajectory dynamics and achieve higher accuracy.

As shown in Figure 3, the figure demonstrates the trend of the mean square error (MSE) of different algorithms with the number of training rounds (epochs) during the training process. In the initial phase, the MSE of all algorithms decreases rapidly. LSTM has a higher initial error but converges quickly after a few epochs. SSA-LSTM has the fastest decrease in MSE in the starting phase and stabilizes at a low level quickly, indicating that SSA-LSTM can find better model configurations at an early stage and converges quickly. In the intermediate stage, the MSE of MLP changes relatively smoothly, but with small fluctuations and slower accuracy improvement. LSTM is affected by noise or training data when capturing nonlinear variations in the data, which results in larger fluctuations in its MSE. In contrast, SSA-LSTM is more adaptive to the data after hyperparameter optimization, significantly reduces the error, and remains stable; thus, SSA-LSTM has the lowest and smoother MSE at this stage. After 100 epochs, all the algorithms tend to be stable. SSA-LSTM has the smallest MSE and almost closes to zero, and has higher prediction accuracy after training. The MSE of MLP and LSTM is still higher than that of SSA-LSTM in the later stage, and the MSE of LSTM is slightly higher than that of MLP. Overall, SSA-LSTM algorithms have the lowest error in all stages and are suitable for subsequent ship prediction, which is suitable for the subsequent prediction of ship routes.

5.2. Performance Evaluation of UAV Dynamic Deployment Algorithms

In this section, we evaluate the performance of the proposed RUDD algorithm through simulation tests. In the simulation experiments, we compare the proposed RUDD algorithm, PPO alone, Deep Deterministic Policy Gradient (DDPG) algorithm, and Random policy (Random) algorithm for the network coverage, the latency of the user to receive the service, and the average energy consumption of the UAVs for completing the service. The main simulation parameters of the RUDD algorithm are shown in Table 1.

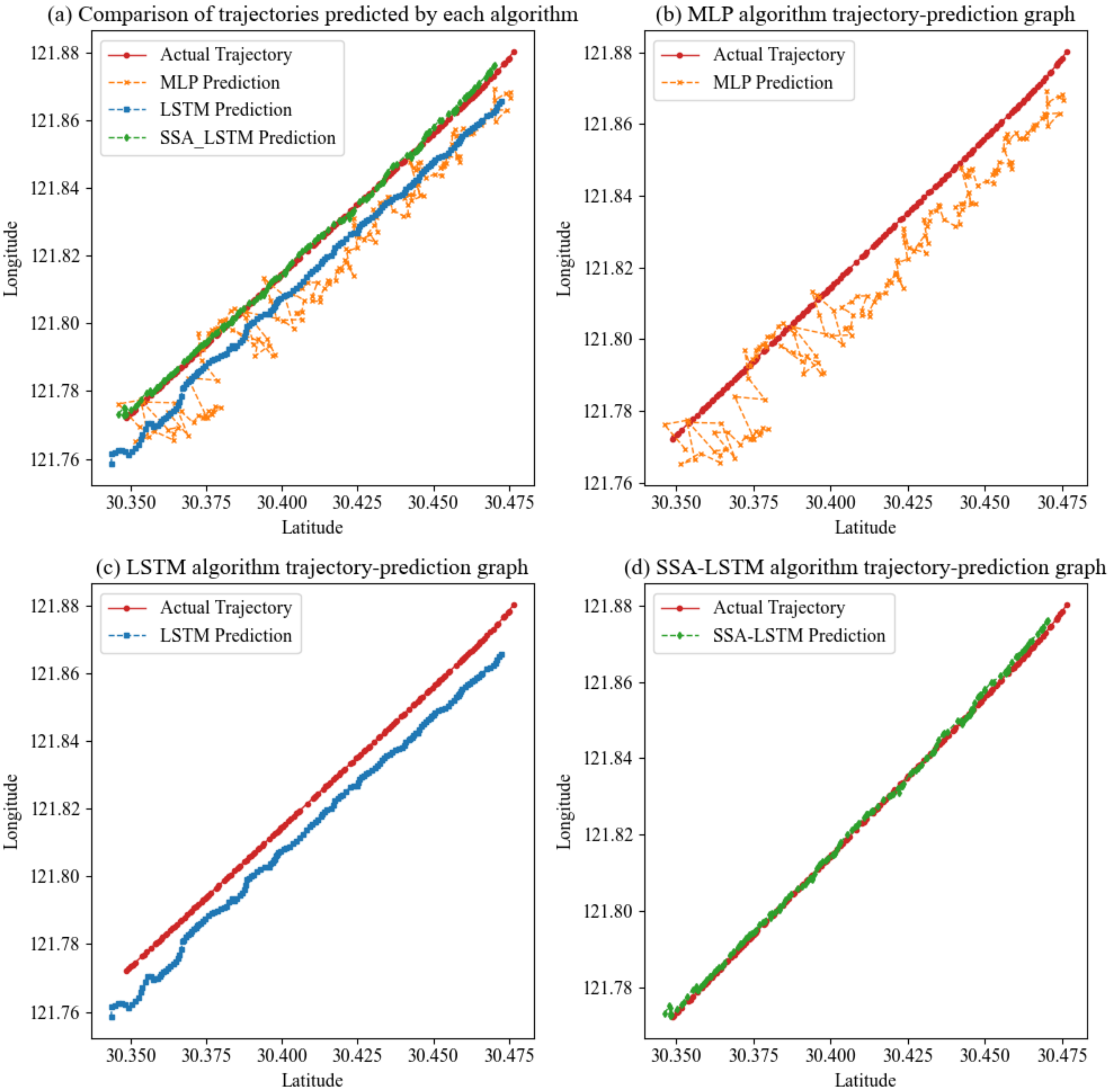

Figure 2. (**a**) Comparison of trajectories predicted by each algorithm; (**b**) MLP algorithm trajectory-prediction graph; (**c**) LSTM algorithm trajectory-prediction graph; (**d**) SSA-LSTM algorithm trajectory-prediction graph.

In this experiment, the parameters we chose include the number of mobile devices, the number of UAVs, the discount factor γ, the decay factor λ, the number of training rounds, the Critic learning rate, and the Actor learning rate in $A^{\pi_{\theta_{\text{old}}}}$. The number of mobile devices will determine their initial distribution and location, which determines the complexity of the task and the system load, and setting it to $[10, 15, 20, 25, 30]$ can test the adaptability and robustness of the algorithm under different densities of user scenarios. The number of drones is fixed at 3, which determines the service capacity and coverage, thus placing restrictions on the computational capacity, energy consumption, and coverage of the whole algorithm. The discount factor γ can determine how much importance the algorithm attaches to future rewards, while the decay factor λ can be used to control the balance between the variance and bias of the dominance estimation, and setting them to 0.95 and 0.9, respectively, can make the algorithm pay more attention to long-term rewards. The algorithm tends to converge at around 200 training rounds and thus is set to 200. The Critic network learning rate can be used to update the value function quickly, and the Actor-network learning rate can be used to avoid the model instability caused by a too fast strategy update. Setting them to 0.02 and 0.01, respectively, can help the smooth optimization of the strategy, which can improve the robustness of the algorithm.

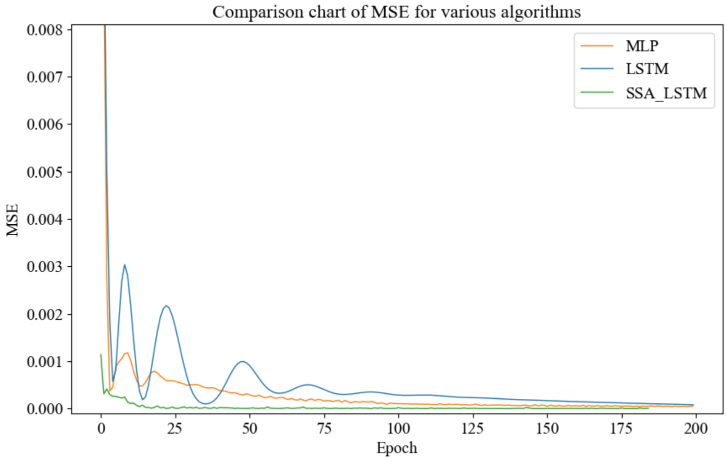

Figure 3. Comparison of the trend of prediction error of each algorithm with the number of training rounds.

Table 1. Main simulation parameters.

System Parameters	Value Range
Number of mobile devices	[10, 15, 20, 25, 30]
Number of UAVs	3
Discount factor γ	0.95
Decay factor λ	0.9
Training episode	200
Critical learning rate	0.02
Actor learning rate	[0.001, 0.002, 0.01, 0.1]

As shown in the comparison of the total reward curves of the four algorithms in Figure 4, the PPO algorithm performed more stably and had relatively higher rewards than the Random and DDPG algorithms. The PPO used a restrained update mechanism, which ensured smoothness during training by limiting the magnitude of each policy update, thus avoiding excessive policy fluctuations. The Random strategy, due to the lack of a learning mechanism, did not perform significantly better as the number of training times increased and thus performed relatively poorly. In this experimental setting, the DDPG's reliance on empirical playback and unconstrained policy updates results in its vulnerability to noise and outdated data during training. As a result, the DDPG algorithm fluctuates and struggles to converge in reward values as the number of training sessions increases. Compared to these three, RUDD adds a prediction algorithm on top of PPO, which enables the UAV to predict the distribution of users in advance, thus optimizing policy updates and making appropriate action choices. As a result, the UAV can formulate the best deployment method before the task execution, which not only ensures coverage but also effectively reduces flight delay and energy consumption. This makes RUDD show strong adaptability and stability during the training process, and thus the reward value after final convergence is the largest. The RUDD algorithm can predict environmental changes more accurately in complex and dynamic marine environments, thus realizing the dynamic deployment of UAVs, and improving the stability and the effectiveness of mission execution under uncertain and dynamic conditions.

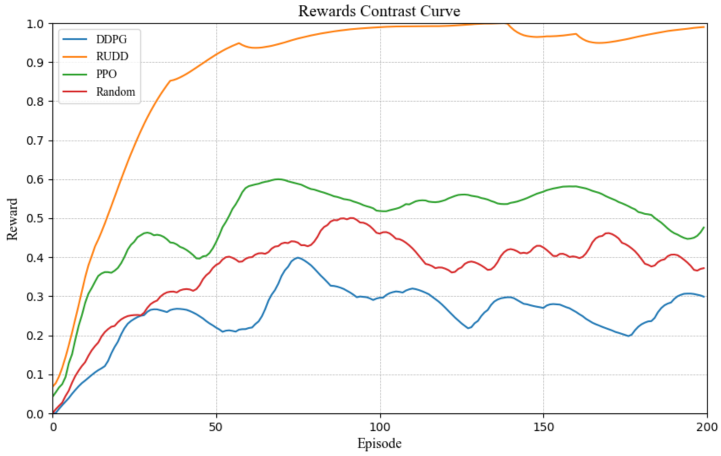

Figure 4. Comparison of total reward values for different models.

As shown in Figure 5, in the same sea area, with the increase in user density, although the UAV can reach the coverage condition more easily, the coverage rate of users also shows a decreasing trend due to the relative increase in the number of users distributed at the edge. The PPO algorithm shows relatively strong stability in the change of coverage under different numbers of users, and its coverage decreases from 95% with 10 users to about 80% with 30 users. Despite the decrease, the performance is better compared to the Random and DDPG algorithms. The PPO algorithm can maintain a high coverage rate more stably with the increase in the number of users and has relatively good adaptive ability. In contrast, the Random algorithm, due to the lack of policy optimization, cannot respond effectively in the face of more users, resulting in a significant drop in its performance. As a result, Random performs poorly when the number of users increases, and its coverage gradually decreases to about 72% with the increase in users. Due to the interference of noise, the DDPG algorithm's policy update is more unstable, and its coverage performance is worse as the number of users increases, dropping from 90% to less than 70%. RUDD adds a prediction algorithm to the PPO, which can optimize the policy in advance, making it able to maintain a higher coverage in the face of the dynamically changing marine environment of the users, and showing great stability and adaptability. As a result, the RUDD algorithm has strong robustness as the coverage rate only drops from about 98% to 85% even with the increase in the number of users.

As shown in Figure 6, within the same sea area, as the user density increases, the UAVs can meet the coverage requirements with relative ease, which in turn leads to a reduction in the total flight delay. This results in a decreasing trend in the average user delay. In different user size scenarios, both DDPG and Random algorithms show a significant decreasing trend as the number of users increases. However, under the constraints of UAV computing power, battery capacity, and coverage area, although the latency of these two algorithms decreases rapidly, due to the limitations of the algorithms themselves, their latency is still higher than that of the RUDD algorithm and the PPO algorithm when the number of users reaches 30. Meanwhile, the PPO algorithm can achieve the best effect of the algorithm more stably by its effective control of the magnitude of policy updates. Based on PPO, we introduce a prediction algorithm, which allows UAVs to quickly make deployment strategies based on the distribution state of users. As a result, the RUDD algorithm has the best performance among all the algorithms, with an average response delay of 1s and the smallest curve variation, which significantly improves the overall execution efficiency and stability. The low-latency performance of the RUDD algorithm ensures that the UAV achieves fast response and stable coverage in maritime communication, which greatly improves the safety and efficiency of maritime operations.

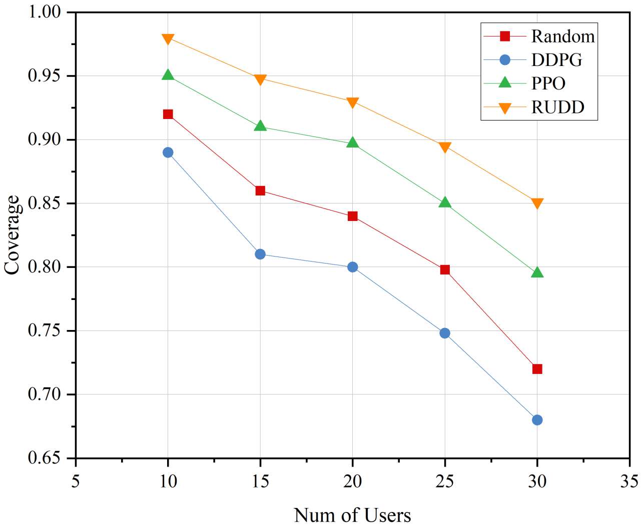

Figure 5. Coverage change graph for different numbers of users.

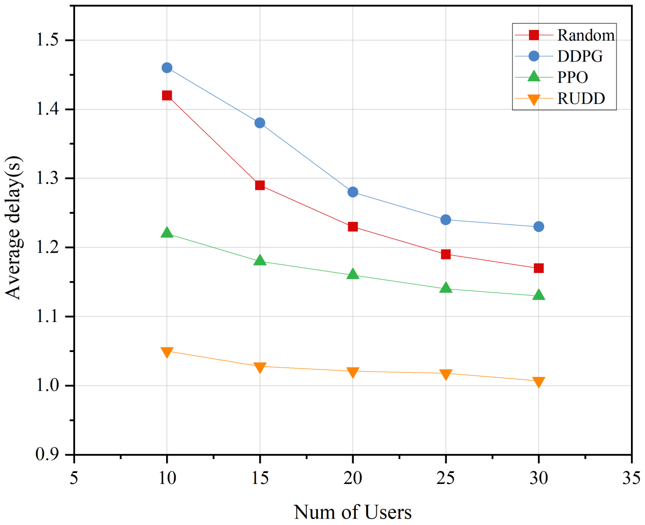

Figure 6. Comparison of the average latency with different numbers of users.

As shown in Figure 7, as the number of users increases, the flight energy consumption and hover energy consumption decrease due to the decrease in the average delay of users. Thus, the average energy consumption of users also shows a decreasing trend. When comparing the average energy consumption under different numbers of users, the RUDD algorithm can adjust the strategy in advance and optimize the energy allocation due to the prediction mechanism, thus effectively reducing the energy consumption while ensuring coverage. As a result, it has the best performance among the four algorithms, with the average energy consumption of users remaining stable at around 710 mAh. In contrast, the PPO algorithm has a relatively stable performance in terms of energy consumption, but due to the lack of the ability to adjust the strategy in advance, its energy consumption is slightly higher than that of RUDD, with an average energy consumption of about 800 mAh at about 30 users. With DDPG and Random algorithms it is difficult to effectively control energy consumption in dynamic environments since the algorithm's strategy is not reasonably limited. Both of them show an obvious decreasing trend, but the uncertainty of strategy selection leads to their energy consumption always being higher. The average user energy consumption of the DDPG and Random algorithms is reduced to 850 mAh and 825 mAh, respectively, when the number of users is 30. Therefore, when applied to offshore scenarios, the RUDD algorithm can efficiently optimize energy utilization, and significantly improve the economy and durability of UAV operations.

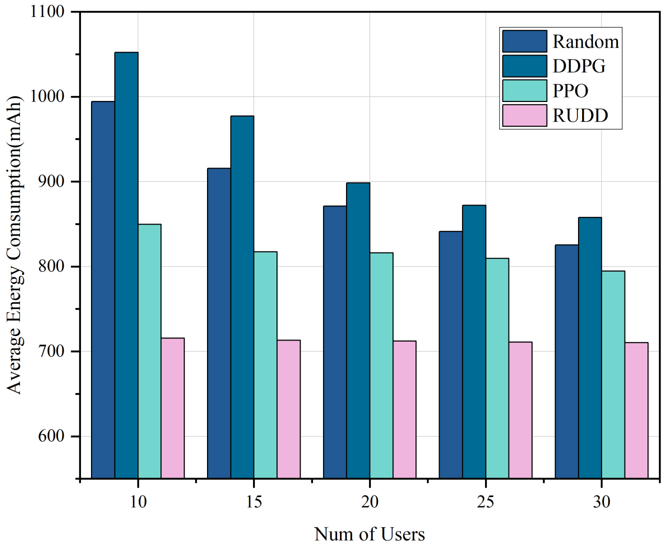

Figure 7. Comparison of average energy consumption for different numbers of users.

6. Conclusions

In this paper, we propose a dynamic deployment scheme for maritime UAVs based on route prediction, aiming to improve the communication coverage of mobile users under the constraints of UAV battery capacity, coverage, and computational resources. With an improved LSTM algorithm, we accurately predict the route distribution of maritime users, thus optimizing the deployment strategy. Combined with the PPO algorithm, this study innovatively utilizes the route-prediction results to enable the agent to formulate UAV-deployment strategies in advance based on the user distribution, thus effectively reducing the latency of task processing. Simulation results show that compared with traditional methods, the RUDD algorithm exhibits high stability in marine scenarios, the average processing delay of the task is successfully reduced to about 1 s, and the communication coverage is improved to more than 95%. This study provides an effective strategy for the deployment of UAV-assisted wireless sensor networks with a wide range of practical applications. Future research will focus on studying more complex marine environments and exploring how to further reduce the amount of algorithmic computation in order to further reduce the total system cost.

Author Contributions: Conceptualization, Y.X. and Y.S.; methodology, Y.X. and Y.S.; software, Y.S.; validation, Y.S.; formal analysis, Y.X.; investigation, Y.S.; resources, Y.X.; data curation, Y.S.; writing—original draft preparation, Y.S.; writing—review and editing, Y.X.; visualization, Y.S.; supervision, Y.X.; funding acquisition, Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Natural Science Foundation of China (62271303)/the Innovation Program of Shanghai Municipal Education Commission of China (2021-01-07-00-10-E00121), and the Natural Science Foundation of Shanghai (20ZR1423200) (author: Yanli Xu).

Data Availability Statement: The data presented in this study are available in Section 5.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Xia, T.; Wang, M.M.; Zhang, J.; Wang, L. Maritime internet of things: Challenges and solutions. *IEEE Wirel. Commun.* **2020**, *27*, 188–196. [CrossRef]
2. Wei, T.; Feng, W.; Chen, Y.; Wang, C.X.; Ge, N.; Lu, J. Hybrid satellite–terrestrial communication networks for the maritime Internet of Things: Key technologies, opportunities, and challenges. *IEEE Internet Things J.* **2021**, *8*, 8910–8934. [CrossRef]
3. Alqurashi, F.S.; Trichili, A.; Saeed, N.; Ooi, B.S.; Alouini, M.S. Maritime communications: A survey on enabling technologies, opportunities, and challenges. *IEEE Internet Things J.* **2022**, *10*, 3525–3547. [CrossRef]
4. Liang, Z.; Dai, Y.; Lyu, L.; Lin, B. Adaptive data collection and offloading in multi-UAV-assisted maritime IoT systems: A deep reinforcement learning approach. *Remote Sens.* **2023**, *15*, 292. [CrossRef]
5. Khan, M.A.; Alzahrani, B.A.; Barnawi, A.; Al-Barakati, A.; Irshad, A.; Chaudhry, S.A. A resource friendly authentication scheme for space–air–ground–sea integrated Maritime Communication Network. *Ocean Eng.* **2022**, *250*, 110894. [CrossRef]
6. d'Orey, P.M.; Gaitán, M.G.; Santos, P.M.; Ribeiro, M.; de Sousa, J.B.; Almeida, L. Assessing Short-range Shore-to-Shore (S2S) and Shore-to-Vessel (S2V) WiFi Communications. *Comput. Netw.* **2024**, *249*, 110505. [CrossRef]
7. Diaz-Vilor, C.; Almasi, M.A.; Abdelhady, A.M.; Celik, A.; Eltawil, A.M.; Jafarkhani, H. Sensing and Communication in UAV Cellular Networks: Design and Optimization. *IEEE Trans. Wirel. Commun.* **2023**, *23*, 5456–5472. [CrossRef]
8. Shen, J.; Bai, X.; Tu, X.; Liu, J. Cost-effective task offloading and trajectory optimization in UAV assisted edge networks with DDPG. *Int. J. Web Inf. Syst.* **2024**, *20*, 494–519. [CrossRef]
9. Gao, Y.; Liu, M.; Yuan, X.; Hu, Y.; Sun, P.; Schmeink, A. Federated deep reinforcement learning based trajectory design for UAV-assisted networks with mobile ground devices. *Sci. Rep.* **2024**, *14*, 22753. [CrossRef]
10. Mohamed, E.M.; Alnakhli, M.; Fouda, M.M. Joint UAV Trajectory Planning and LEO-Sat Selection in SAGIN. *IEEE Open J. Commun. Soc.* **2024**, *5*, 1624–1638. [CrossRef]
11. Yang, Y.; Liu, Y.; Li, G.; Zhang, Z.; Liu, Y. Harnessing the power of Machine learning for AIS Data-Driven maritime Research: A comprehensive review. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *183*, 103426. [CrossRef]
12. Yuan, X.; Wang, J.; Zhao, G.; Wang, H. Comprehensive Study on Optimizing Inland Waterway Vessel Routes Using AIS Data. *J. Mar. Sci. Eng.* **2024**, *12*, 1775. [CrossRef]
13. Wang, P.; Di, B.; Song, L. Multi-layer LEO satellite constellation design for seamless global coverage. In Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 7–11 December 2021; pp. 1–6.
14. Chen, G.; Wu, S.; Deng, Y.; Jiao, J.; Zhang, Q. VLEO Satellite Constellation Design for Regional Aviation and Marine Coverage. *IEEE Trans. Netw. Sci. Eng.* **2023**, *11*, 1188–1201. [CrossRef]
15. Talgat, A.; Kishk, M.A.; Alouini, M.S. Stochastic geometry-based analysis of LEO satellite communication systems. *IEEE Commun. Lett.* **2020**, *25*, 2458–2462. [CrossRef]
16. Xie, H.; Yang, D.; Xiao, L.; Lyu, J. Connectivity-aware 3D UAV path design with deep reinforcement learning. *IEEE Trans. Veh. Technol.* **2021**, *70*, 13022–13034. [CrossRef]
17. Peng, J.; Tang, W.; Zhang, H. Directional antennas modeling and coverage analysis of UAV-assisted networks. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 2175–2179. [CrossRef]
18. Liu, X.; Wang, X.; Jia, J.; Lv, J.; Bartolini, N. Distributed deployment in UAV-assisted networks for a long-lasting communication coverage. *IEEE Syst. J.* **2021**, *16*, 4130–4138. [CrossRef]
19. Zhu, X.; Zhai, L.; Li, N.; Li, Y.; Yang, F. Multi-objective Deployment Optimization of UAVs for Energy-Efficient Wireless Coverage. *IEEE Trans. Commun.* **2024**, *72*, 3587–3601. [CrossRef]
20. Fu, H.; Wang, J.; Chen, J.; Ren, P.; Zhang, Z.; Zhao, G. Dense Multi-Agent Reinforcement Learning Aided Multi-UAV Information Coverage for Vehicular Networks. *IEEE Internet Things J.* **2024**, *11*, 21274–21286. [CrossRef]
21. Bezas, K.; Tsoumanis, G.; Oikonomou, K. A coverage path planning algorithm for self-organizing drone swarms. In Proceedings of the 2021 International Balkan Conference on Communications and Networking (BalkanCom), Novi Sad, Serbia, 20–22 September 2021; pp. 122–126.
22. Xu, Z. Application research of tethered UAV platform in marine emergency communication network. *J. Web Eng.* **2021**, *20*, 491–511. [CrossRef]
23. Huang, H.; Savkin, A.V. Deployment of heterogeneous UAV base stations for optimal quality of coverage. *IEEE Internet Things J.* **2022**, *9*, 16429–16437. [CrossRef]
24. Xu, J.; Kishk, M.A.; Alouini, M.S. Space-air-ground-sea integrated networks: Modeling and coverage analysis. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 6298–6313. [CrossRef]
25. Fang, X.; Feng, W.; Wang, Y.; Chen, Y.; Ge, N.; Ding, Z.; Zhu, H. NOMA-based hybrid satellite–UAV–terrestrial networks for 6G maritime coverage. *IEEE Trans. Wirel. Commun.* **2022**, *22*, 138–152. [CrossRef]
26. Li, X.; Feng, W.; Chen, Y.; Wang, C.X.; Ge, N. Maritime coverage enhancement using UAVs coordinated with hybrid satellite–terrestrial networks. *IEEE Trans. Commun.* **2020**, *68*, 2355–2369. [CrossRef]
27. Wang, Y.; Feng, W.; Wang, J.; Quek, T.Q. Hybrid satellite–UAV–terrestrial networks for 6G ubiquitous coverage: A maritime communications perspective. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3475–3490. [CrossRef]
28. Shin, G.H.; Yang, H. Vessel Trajectory Prediction at Inner Harbor Based on Deep Learning Using AIS Data. *J. Mar. Sci. Eng.* **2024**, *12*, 1739. [CrossRef]

29. Xiao, Y.; Li, X.; Yao, W.; Chen, J.; Hu, Y. Bidirectional data-driven trajectory prediction for intelligent maritime traffic. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 1773–1785. [CrossRef]
30. Yang, F.; He, C.; Liu, Y.; Zeng, A.; Hu, L. Vessel Trajectory Prediction Based on Automatic Identification System Data: Multi-Gated Attention Encoder Decoder Network. *J. Mar. Sci. Eng.* **2024**, *12*, 1695. [CrossRef]
31. Li, X.; Liu, C.; Li, J.; Zhao, L.; Du, Z. Advancing ship trajectory prediction: Integrating deep learning with enhanced reference trajectory correction techniques. *Ocean Eng.* **2024**, *311*, 118880. [CrossRef]
32. Li, H.; Xing, W.; Jiao, H.; Yang, Z.; Li, Y. Deep bi-directional information-empowered ship trajectory prediction for maritime autonomous surface ships. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *181*, 103367. [CrossRef]
33. Zhang, Y.; Yuan, X.; Li, M.; Zhao, G.; Wang, H. Multi-Density Adaptive Trajectory Clustering Algorithm For Ships Based on AIS Data. *IEEE Access* **2023**, *11*, 108198–108210. [CrossRef]
34. Alam, M.M.; Spadon, G.; Etemad, M.; Torgo, L.; Milios, E. Enhancing short-term vessel trajectory prediction with clustering for heterogeneous and multi-modal movement patterns. *Ocean Eng.* **2024**, *308*, 118303. [CrossRef]
35. Zhang, Y.; Han, Z.; Zhou, X.; Li, B.; Zhang, L.; Zhen, E.; Wang, S.; Zhao, Z.; Guo, Z. METO-S2S: A S2S based vessel trajectory-prediction method with Multiple-semantic Encoder and Type-Oriented Decoder. *Ocean Eng.* **2023**, *277*, 114248. [CrossRef]
36. Luan, Z.; Jia, H.; Wang, P.; Jia, R.; Chen, B. Joint UAVs' load balancing and UEs' data rate fairness optimization by diffusion UAV deployment algorithm in multi-UAV networks. *Entropy* **2021**, *23*, 1470. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.