Computer Communications 213 (2024) 260–270

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Heterogeneous temporal graph powered DRL algorithm for channel allocation in Maritime IoT Systems

Zongwang Li, Zhuochen Xie, Xiaohe He, Xuwen Liang *

Innovation Academy for Microsatellites of Chinese Academy of Sciences, No. 1 Xueyang Road, Pudong District, Shanghai, 201304, China
University of Chinese Academy of Sciences, No. 1 Yanqihu East Rd, Huairou District, Beijing, 101408, China

ARTICLE INFO

Keywords:
Maritime IoT
Heterogeneous temporal graph
Deep reinforcement learning

ABSTRACT

In actual maritime Internet of Things systems, the communication environment is characterized by its time-varying nature and the presence of highly heterogeneous network structures. Those attributes present considerable challenges in devising resource allocation strategies. Given the limited availability of frequency resources, designing a reasonable and flexible channel allocation strategy (CAS) is the primary task for meeting diverse and dynamic communication demands. In this paper, a heterogeneous temporal graph powered deep reinforcement learning algorithm is proposed to optimize the CAS to maximize the channel efficiency in a real-world maritime Internet of Things system. Specifically, we build relation-based heterogeneous edges to connect different types of terminal nodes and adopt a time encoding technology to capture the dynamic evolution of communication scenarios over time. The memory and public mailbox modules are constructed as the implementation entities of the information aggregation method based on the attention mechanism. In addition, we develop a corresponding heterogeneous temporal neural network to estimate the real-time resource requirements of target terminals, and subsequently learn the optimal CAS based on the deep reinforcement learning algorithm from the perspective of maximizing the cumulative channel efficiency. Simulation results prove that the proposed algorithm significantly outperforms the other state-of-the-art algorithms in terms of the channel efficiency and generalization ability.

1. Introduction

The oceans cover more than 70 percent of the Earth's surface, and international maritime shipping is responsible for the carriage of about 90 percent of world trade [1]. In an effort to modernize the maritime information and communication infrastructure, the United Nations' International Maritime Organization put forward a maritime Internet of Things (IoT) concept, under the name e-Navigation [2]. This concept is further extended and technically formalized to a fully-fledged maritime IoT framework, under which all vessels and maritime equipment, i.e., the maritime "things" are interconnected through a unified machine-type communication (MTC) system for undisrupted maritime services worldwide [3].

As with any other IoT application, channel allocation strategy (CAS) is one of the critical technology of Maritime IoT due to the extremely scarce wireless communication resources. Complex environments and heterogeneous networks greatly challenge the efficiency and adaptability of CAS. On the one hand, the distribution of maritime traffic is extremely uneven in both spatial and temporal dimension. On the other hand, the steady growth in oceanic activities has led to diverse maritime IoT applications and services, ranging from basic ship location reporting to intricate marine environmental monitoring and beyond. A maritime IoT system is expected to offer amorphous services that adapt to a wide variety of maritime IoT specific needs, and match changing demands. Consequently, it is difficult to make appropriate decisions on how to utilize the resources under such conditions. With the recent burgeoning application of artificial intelligence (AI) in many fields, an AI-empowered autonomous network for maritime IoT is envisioned as a promising solution [4]. Network modeling is a fundamental component for efficient control and management of communication networks, which can be used for autonomous network control by pairing the model with an automatic optimization algorithm (e.g., local search, reinforcement learning) [5]. Although some studies have used deep learning algorithms based on popular neural networks (e.g., recurrent neural networks) to tackle data prediction and estimation problems in maritime IoT [6,7], these methods are ill-suited for network modeling.

[*] Corresponding author.
E-mail address: 18217631362@163.com (X. Liang).

https://doi.org/10.1016/j.comcom.2023.11.005
Received 29 April 2023; Received in revised form 24 September 2023; Accepted 5 November 2023
Available online 7 November 2023

Maritime IoT networks encompass multifaceted relational information, such as topology, routing, terminal connections, etc. The natural way to represent relational information is in the form of a graph, that is, as a set of elements connected according to their relationships. Graph-structured data are non-Euclidean data [8], which is beyond the learning ability of non-graph neural network models such as fully connected multi-layer perceptrons (MLPs), convolutional neural networks (CNNs), etc. Driven by the graph-structured data, graph neural networks (GNNs) [9] emerge as a more suitable choice, as they can automatically learn a condensed representation of each node in the network that incorporates the information about the node, its neighbors, and their inter-connecting topology and support relational reasoning and combinatorial generalization [10]. Besides, as aforementioned, maritime IoT systems exhibit unevenly distributed and dynamic traffic patterns. In such scenarios, non-graph neural networks are poorly generalized when the system settings in the test datasets, such as the number of terminals, are different from those in the training datasets [11]. In contrast, GNNs are able to capture the complex relationships among topology, routing, and traffic in a network, and generalizes trained NN parameters to any topology, routing scheme, and variable traffic intensity [12]. Therefore, from the perspective of deep learning, maritime IoT system relies on graphs as fundamental elements to represent networks to solve a plethora number of control and optimization problems. This ultimately calls for applying deep learning approaches that are more suitable for graph-structured data.

In light of the issues above, we designed a heterogeneous temporal graph (HTG) model to represent the real-world maritime IoT network, and developed a corresponding HTG neural network (HTGNN) to estimate the traffic of the target terminal. Moreover, combining with deep reinforcement learning (DRL) algorithm, an intelligent CAS is proposed to maximize the utilization of frequency resources. The following are our main contributions:

- A maritime IoT network modeling approach based on HTG is proposed for the first time to couple the heterogeneous and time-varying traffic information of the maritime IoT into a unified graph-structured data.
- A multi-relations attention-based HTGNN is designed to estimate the real-time resource demand of maritime IoT terminals. We adopt a time encoding method to represent the time factor, and then the influence of neighbor nodes associated with diverse relations is converted into a set of importance coefficients to aggregate neighbor information from both time dimension and relation dimension.
- The channel allocation problem in an actual maritime IoT system is transformed into an Markov Decision Process (MDP), and we propose a DRL based CAS to maximize the channel efficiency according to the resource demand estimation of target terminal and the relevant communication environment information.
- To evaluate the performance of our proposed algorithm, we conduct a series of simulations on multiple datasets with different traffic intensities. The simulation results show that the proposed algorithm outperforms the state-of-the-art methods in terms of channel efficiency, convergence and generalization ability.

The remainder of this paper is as follows. We review related work on GNN theory and GNN based algorithm for resources allocation of communication networks in Section 2. In Section 3, an optimization model for channel allocation in practical maritime IoT systems is established and analyzed. In Section 4, a HTG model based on maritime IoT systems is constructed, and a corresponding HTGNN is proposed to estimate terminal's resource demands. Building upon this, in Section 5, a DRL algorithm-based CAS is introduced with the aim of obtaining the optimal solution for the constructed optimization model. Section 6 shows the simulation results of performance evaluation. We conclude the paper in Section 7 with a brief summary.

2. Related work

2.1. GNN theory

Due to impressive performance of GNN in analyzing non-Euclidean data, GNN-based deep learning methods have become a research hotspot in recent years [9]. As an extension of CNN in the field of graph data, graph convolutional network [13,14] aggregates neighbor information according to the connection relationship of the graph structure. On this basis, graph attention network [15] introduces the attention mechanism in the aggregation process, and learns the importance of neighbor nodes through a self-attention strategy. Moreover, the latest GNN-based researches mainly focuses on dealing with the heterogeneity and dynamics of graph data.

Considering that the graph-structure data of real-world networks usually have diverse types of nodes and edges, a variety of heterogeneous graph neural network models have been proposed to explore the representation learning in heterogeneous graphs. A multi-level attention mechanism is proposed in [16], which extracts and aggregates the information of neighbor nodes based on each edge type, and then fuses the aggregation results across different edges. The authors in [17] adopt different feature extraction networks for different node features. In [18], Ziniu Hu et al. design node- and edge-type dependent parameters to characterize the heterogeneous attention over each edge to maintain dedicated representations for different types of nodes and edges. The classification results of proposed model on Open Academic Graph datasets outperform those of the above two models.

The above models are all based on static graphs. But in real world, the entities modeled as graph present different temporal dynamic in node features and relations. The extra time dimension brings temporal information to the graph's representation and increases the difficulty of analysis as well. Dynamic graph can be modeled as either Discrete Time Dynamic Graph (DTDG) or Continuous Time Dynamic Graph (CTDG) based on how the temporal information is expressed regarding to the evolution of dynamic graph [19]. DTDG is a list of snapshots and each of them keeps the graph status at a certain moment. Meanwhile, CTDG can be viewed as a stream of graph updating event. DTDG storage model relies on appropriate sampling frequency to reduce the loss of important temporal information [20], which is not suitable for communication networks with a large number of random events. Beside, each snapshot is derived from the traversal of all graph nodes, which is not practical in the Maritime IoT system. Therefore, we mainly review the dynamic graph model based on the CTDG. A temporal graph attention (TGAT) layer is proposed in [21], the authors use the self-attention mechanism as building block and develop a novel functional time encoding technique based on the classical Bochner's theorem from harmonic analysis to represent the node embeddings as functions of time. [22,23] improve TGAT-based model from the perspective of general framework and inference time consumption, respectively.

In addition, some researches [24,25] have tried to solve the heterogeneity problem of dynamic graphs. However, these methods are DTDG-based and cannot fully learn the time information in heterogeneous graphs. To the best of our knowledge, existing literature lacks the research on CTDG-based heterogeneous temporal GNN models.

2.2. GNN based algorithm for resources allocation

As mentioned earlier, compared with traditional neural networks, GNNs have more advanced performance and generalization ability on the representation learning of graph-structured data by learning the relevant information hidden in the topological graph structure [5,10]. Therefore, researchers have proposed various deep learning approaches base on GNNs to resolve the resource allocation problem of communication networks. The authors in [26] propose a GNN-based supervised learning architecture that models wireless networks as directed graphs, improving the generalization and robustness of the algorithm. A power

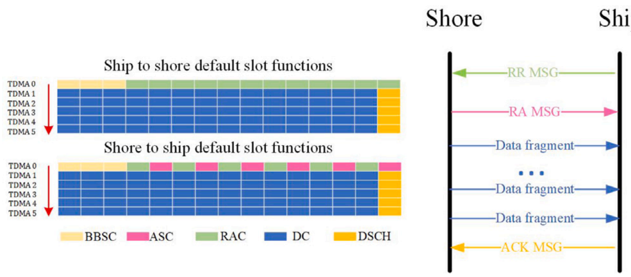


Fig. 1. Slot functions and data transfer protocol.

allocation algorithm based on heterogeneous GNN is designed in [27] to match the policy with the properties of networks. Compared with homogeneous GNN, the proposed algorithm reduces the number of samples required for training. Zhang et al. models the heterogeneous ultra-dense network as a heterogeneous bipartite graph in [28], and propose a corresponding heterogeneous bipartite GNN. Combining with data-driven and model-driven learning, the proposed method has better performance and better computational efficiency than traditional algorithms. However, the above algorithms learn the optimal strategy from the perspective of maximizing the current benefit, without considering the impact of current allocation on future environment.

In comparison, DRL algorithm is able to learn the internal correlation between sequential decisions by interacting with the environment to find a strategy for maximizing cumulative benefits [29]. [30] applies GCN-based Deep Q-network to learn the channel allocation strategy in wireless LANs. Compared with the method of real-time reward maximization, the proposed algorithm improves the throughput of the system. A dynamic GNN-based algorithm is proposed in [31] to solve the time-varying problem of the actual environment in the DRL framework, which achieves better performance than static graphs. However, further work is required to integrate the heterogeneity and dynamics of graph-structure data in DRL with GNNs [32], which is crucial for the network modeling of real maritime IoT systems.

To sum up, no matter from the perspective of GNN theory or from the perspective of application based on GNN, there is a lack of an effective method suitable for network modeling of real maritime IoT systems. Therefore, we design a CTDG-based HTG model to help fill this gap. Meanwhile, a corresponding HTGNN is developed to analyze the constructed HGT. We apply this model in a CAS based on DRL algorithm to estimate the resource requirements of target terminals, which eventually improves the channel utilization efficiency and generalization ability of the strategy. The implementation details are described in Sections 3 and 4.

3. System models

In this section, we transform the channel allocation problem into an optimization problem based on a real-world maritime IoT system, known as VHF Data Exchange System (VDES) [33–35], and analyze the challenges and solution ideas for this problem.

3.1. Data transfer procedure

CAS is one of the core steps in the data transmission protocol. To provide a clearer description of the channel allocation problem, we first introduce relevant details of the data transmission protocol. Taking VHF data exchange-terrestrial (VDE-TER) in VDES [35] as the communication background, we consider a port scenario with dense IoT devices, where the centralized CAS is employed to avoid the collisions in data channel (DC). The slot functions and data transfer protocol details of ship to shore are shown in Fig. 1. VDE-TER is a typical time division multiple access (TDMA) system and 6 TDMA channels constitute a TDMA frame. Before initiating data transmission, the ship station must monitor the bulletin board signalling channel (BBSC) to obtain the necessary channel configuration parameters, such as time slot mapping configuration. Once the ship station acquires those information, it can send resource request message (RR MSG) in random access channel (RAC) to request communication resource. If two or more ship stations choose the same RAC slot, collision will occur. Otherwise, we consider that the shore station can successfully receive this RR message. After receiving the RR MSG, the shore station will determine the number of DC slots allocated to this ship station (herein called target terminal) according to CAS, and broadcast it on the announcement signalling channel (ASC) via the resource allocation message (RA MSG). The target terminal then divides the data to be transmitted into corresponding VDE payloads according to the number of allocated DC slots. Finally, the data transmission result will be confirmed by an acknowledgment message (ACK MSG) in the corresponding data signalling channel (DSCH). It is worth noting that when the ship station has remaining data not to be sent, it will request more DC slots in the last data fragment, which does not require an extra random access procedure.

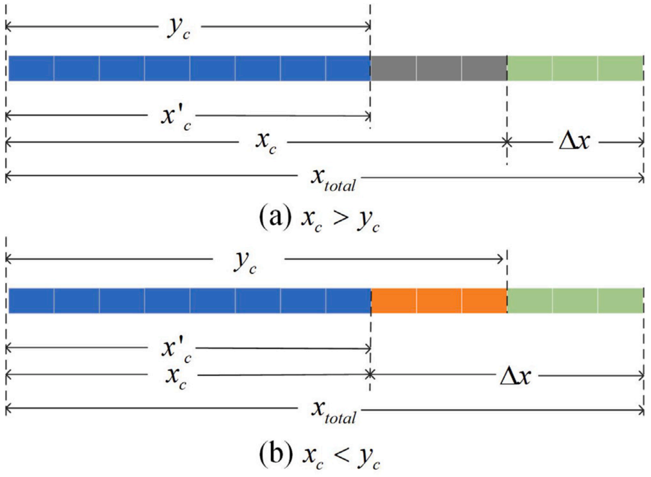


Fig. 2. Schematic diagram of slot utilization.

3.2. Problem modeling

Assuming that the total number of DCs in the period from 0 to T is C. As illustrated in Fig. 2, each DC has x_{total} slots, the number of slots allocated to the target terminal by the shore station and required by this terminal in the cth DC are denoted by x_c and y_c, respectively. $\Delta x = x_{total} - x_c$ denotes the number of unassigned slots which can be used as RAC to improve the probability of successful access for RR MSG. Constrained by the extremely scarce communication resources, efficiency is the only vital means to maximize the system capacity [1], we design the efficiency gain function of the cth DC as follow:

$$f(x_c, x'_c) = \alpha \frac{x'_c}{x_{total}} + \beta(1 - \frac{x_c}{x_{total}}), \tag{1}$$

where, $x'_c = min(y_c, x_c)$ represents the number of slots actually used in the cth DC, α and β are efficiency gain coefficients of the DC and RAC, respectively, $\alpha > \beta$. The optimal model of channel efficiency in the period from 0 to T can be formulated as:

$$\begin{aligned} &\max_{x_c} \sum_{c=1}^{C} f(x_c, x'_c) \\ &s.t. \quad 0 \le x'_c \le x_c \le x_{total}, c = 1, 2, \dots, C \quad . \end{aligned} \tag{2}$$

Obviously, if y_c is a constant, (2) obtains the maximum value $\frac{\alpha-\beta}{x_{total}} \sum_{c=1}^{C} y_c + \beta C$ when $x_c = y_c$. However, in a scenario with dense

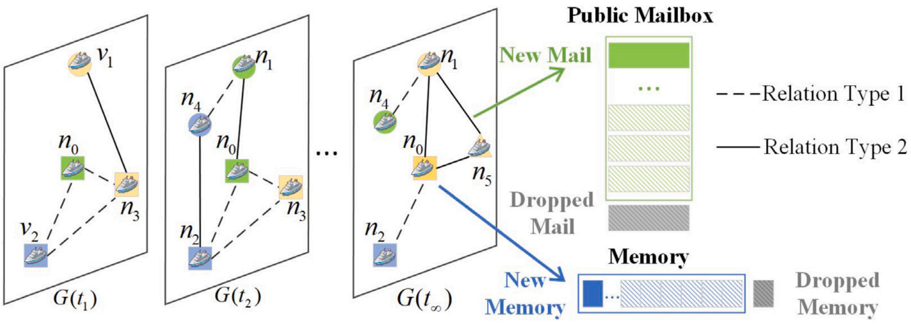


Fig. 3. HTG schematic diagram of the communication scene within the service range of the shore station.

terminals, the number of ship stations successfully accessing the network decreases sharply due to the collision of RR MSG, which results in a lower received total resource requirements $\sum_{c=1}^{C} y_c$. Increasing the number of RAC slots can effectively alleviate the collision problem. To put it another way, there is an implicit positive correlation between $\sum_{c=1}^{C} y_c$ and Δx. The main challenges of CAS can be analyzed as follow:

When $x_c > y_c$, as show in Fig. 2(a), this allocation will cause a waste of several DC slots, and those DC slots could have been used as RAC. Such inefficient utilization aggravates the collision of RR MSG, which further leads to the loss of system capacity since some channels may be dataless even though the traffic load is heavy. Appropriately reducing x_c is a potential solution, but it may result in a loss of current channel efficiency as shown in Fig. 2(b). Therefore, the shore station needs to design a flexible CAS to determine the appropriate number of DC slots to maximize the total channel efficiency.

Obviously, in this context, the primary task is to accurately estimate y_c, since the shore station cannot directly obtain the relevant information from RR MSG [35]. y_c is associated with various factors, such as the ship type and the ship status. Traffic load intensity will also affects it in the case where retry mechanism is allowed. The diversity of terminals and the dynamics of the communication environment arouse great challenges to the estimation task.

In summary, in order to maximize long-term channel efficiency, when allocating DC slots, it is advisable to increase the number of RAC slots as much as possible while guaranteeing the current channel efficiency. This necessitates the accurate estimation of y_c. To address the above problems, we will introduce the estimation approach of y_c and the intelligent CAS in Sections 4 and 5, respectively.

4. Resource demand estimation

In order to estimate the resource requirement of the target terminal as accurately as possible, we propose an estimation algorithm based on HTGNN model in this section. To be clear, both the construction of HTG and the estimation of resource requirements are performed by the shore station.

4.1. HTG model

The resource demands of terminals are closely related to the characteristics of communication networks. Given the heterogeneity and dynamics of real-world maritime IoT systems, we will construct the graph model in terms of both topological relationships and temporal variations. Specifically, the communication scenario within the service area of the shore station at time t can be represented as a HTG: $G(t) = \{N(t), E(t), \mathcal{A}(t), \mathcal{R}(t)\}$. Note that t is the time when the RR MSG is successfully received, so the dynamic graph set of $G(t)$ is a graph-structured data based on CTDG, which means that no time information is lost. It will be further simplified later. Each node (terminal) $n \in N(t)$ and each edge $e \in E(t)$ are mapped to the corresponding type according to the type mapping functions $\tau(n) : N(t) \to \mathcal{A}(t)$ and $\phi(e) : E(t) \to \mathcal{R}(t)$ respectively. $\mathcal{A}(t)$ and $\mathcal{R}(t)$ respectively represent the set of node types and the set of edge types at time t, $\mathcal{A}(t)|+|\mathcal{R}(t)| > 2$ in the heterogeneous graph, where $\mathcal{A} = \bigcup_{k=1}^{\infty} \mathcal{A}(t_k)$ and $\mathcal{R} = \bigcup_{k=1}^{\infty} \mathcal{R}(t_k)$.

Define node embedding $h_i(t)$ as the estimated value of y_c, where i denotes the index of the target terminal. In order to better apply the embedding results to downstream tasks, $h_i(t)$ is extended to d_e dimensions. The constructed HTG is shown in Fig. 3, and the details of the construction of its main modules are described as follows:

(1) *Relation-based edge*: Edges are used to represent the topological relationships between graph nodes. Assuming the set of terminal feature types related to y_c is denoted as $\{\Omega_1, \Omega_2, \dots\}$, classify the elements in the feature value domain of each feature Ω, with the resulting classification denoted as $\{\Psi_1^{\Omega}, \Psi_2^{\Omega}, \dots\}$. Establish connections between terminals with feature values based on relationship Ω and belonging to the same category Ψ, defining those connections as edges based on relationship Ω and belonging to category Ψ. It is worth noting that edge types correspond one-to-one with relationship types, i.e., $\mathcal{R} = \{\Omega_1, \Omega_2, \dots\}$. At time t, the edge between node i and node j, based on relationship Ω and belonging to category Ψ, can be represented as $e_{ij}^{\Omega_\Psi}(t)$. Define the category mapping function $\phi_1(e^{\Omega_\Psi}) = \Psi$ and the relationship mapping function $\phi_2(\Psi) = \Omega$, then the type mapping function can be further represented as $\phi(e^{\Omega_\Psi}) = \phi_2(\phi_1(e^{\Omega_\Psi}))$. The edge types set of node i is represented as $\mathcal{R}_i(t) = \{\Omega | e_{ij}^{\Omega}(t) \in E(t)\}$.

(2) *Memory*: Given that the purpose of constructing the HTG is to estimate the resource requirements of target terminal, therefore, we build the Memory module to record the historical information of terminal's resource requirements and update it with a first-in–first-out (FIFO) queue data structure. At time t, Memory of the terminal i can be represented as $M_i(t) = [m_i(t_m^1), m_i(t_m^2), \dots, m_i(t_m^{L_m})]$, $M_i(t) \in \mathbb{R}^{1\times L_m}$, where $m_i(t_m)$ is equal to the x_c' of terminal i at time t_m, which can be obtained after data transmission, $t_m^1 < t_m^2 < \cdots < t_m^{L_m} < t$, L_m is the length of the Memory.

(3) *Public Mailbox*: The resource demands of neighboring nodes reflect the traffic intensity in the current network, and judiciously leveraging this information helps enhance the accuracy of the target terminal's resource requirement estimation. In order to reduce the complexity of querying the information of neighbor nodes, we share this information by maintaining a Public Mailbox (PM) module. During real-time decision-making, the shore station can directly read the mail in the relevant mailbox without polling all the neighbor nodes of the target terminal. PM based on edge e at time t can be expressed as $P_\Psi(t) = [p_\Psi(t_p^1), p_\Psi(t_p^2), \dots, p_\Psi(t_p^{L_p})]^T$, where $P_\Psi(t) \in \mathbb{R}^{L_p\times d_p}$, $t_p^1 < t_p^2 < \cdots < t_p^{L_p} < t$, d_p denotes the dimension of $p(t)$, L_p denotes the mailbox capacity. $p(t)$ can be simply set as the identify result of the node embedding, which means that when $\phi_1(e_{ij}(t)) = \Psi$, $p_\Psi(t) = h_i(t)$. Similar to Memory module, the PM module is updated by FIFO.

Based on the established HTG model, the dynamic graph set $\mathcal{G}$ can be simplified as an event stream of RR MSG arrivals: $\mathcal{G} = \{g(t_1), g(t_2), \dots, g(t_\infty)\}$, where $g(t) = [M_i(t), \{P_\Psi(t) | \phi_1(e_{ij}(t)) = \Psi\}]$. This will significantly improve the feasibility of HTG, since it does not need

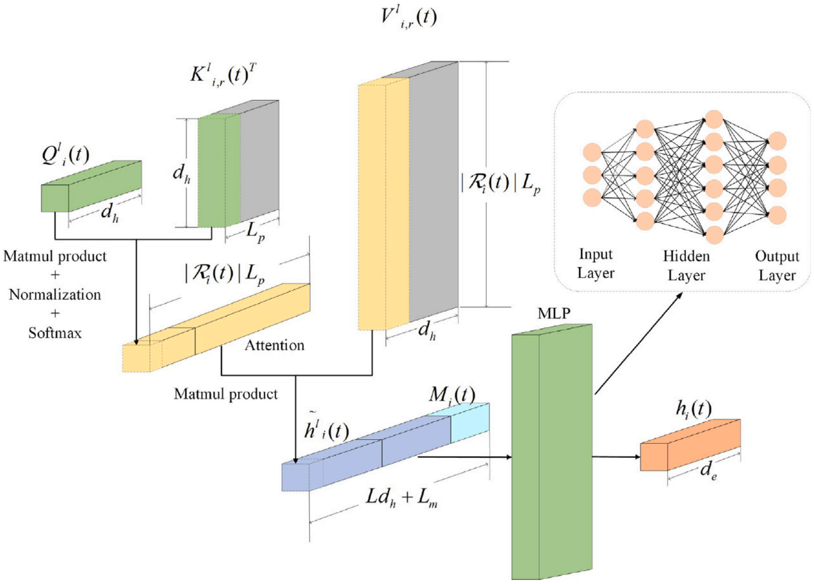


Fig. 4. Diagram of information fusion network structure.

to query all nodes at each time t. Furthermore, the estimation task of y_c can be expressed as follow:

$$h_i(t) = \mathcal{H}(M_i(t), \{P_\Psi(t)|\phi_1(e_{ij}(t)) = \Psi\}), \tag{3}$$

where $\mathcal{H}(\cdot)$ is the estimation function.

4.2. Estimation algorithm based on HTGNN

In this subsection, we further elaborate on the construction details of the estimation function $\mathcal{H}(\cdot)$. In the constructed HTG, edges based on different relationships may contain varying semantic information, implying that the influence of neighbor nodes varies with relation type. Inspired by [18], we propose an HTGNN based on multi-relations-multi-heads (MRMH) attention mechanism to aggregate information of heterogeneous neighbor nodes and estimation function $\mathcal{H}(\cdot)$ can be reformulated as:

$$h_i(t) = Attn_{MRMH}(M_i(t), \{P_\Psi(t)|\phi_1(e_{ij}(t)) = \Psi\}). \tag{4}$$

The implementation of $Attn_{MRMH}$ includes the following three steps:

(1) *Time encoding*: The evolving nature of communication scenarios requires handling new terminals as well as capturing temporal patterns. Moreover, terminal and topological features can be temporal as well, whose patterns the node embeddings should also capture. Thus, we need to encode the temporal information in HTG. Based on Bochner's Theorem and the Monte Carlo integral approximation, we use the method in [21] to define the time encoding function mapping relationship $t \rightarrow \Phi_{d_t}(t)$ as follows:

$$\Phi_{d_t}(t) = \sqrt{\frac{1}{d_t}}[\cos(\omega_1 t), \sin(\omega_1 t), \dots, \cos(\omega_{d_t} t), \sin(\omega_{d_t} t)], \tag{5}$$

where d_t is the number of sampling points. We characterize the influence of temporal factor by encoding the corresponding relative time, which can be denoted as:

$$\widetilde{M}_i(t) = [m_i(t^1_m), \dots, m_i(t^{L_m}_m)]\|\Phi_{d_t}(0), \tag{6}$$

$$\widetilde{P_\Psi}(t) = [p_\Psi(t^1_p)\|\Phi_{d_t}(t - t^1_p), \dots, p_\Psi(t^{L_p}_p)\|\Phi_{d_t}(t - t^{L_p}_p)]^T, \tag{7}$$

where $\|$ represents the concatenation operation. $\widetilde{M}_i(t) \in \mathbb{R}^{1\times(L_m+d_t)}$ and $\widetilde{P_\Psi}(t) \in \mathbb{R}^{L_p\times(d_e+d_t)}$ denote the results of the Memory and the PM fused with time encoding, respectively.

(2) *Information extraction*: In order to obtain sufficient expressive power to transform the input features into higher-level features, at least one learnable linear transformation is required. In particular, based on the established HTG, we adopt different feature extraction matrices to extract the information in the PM of different relations. The process of information extraction can be formulated as:

$$\begin{aligned} Q_i(t) &= \widetilde{M}_i(t)W_q \\ K_{i,r}(t) &= \widetilde{P_\Psi}(t)W_{k,\Omega}|_{\Omega=\phi_2(\Psi)} \\ V_i(t) &= \mathop{\|}_{\Omega\in\mathcal{R}_i(t)} \widetilde{P_\Psi}(t)W_{v,\Omega}|_{\Omega=\phi_2(\Psi)} \end{aligned}, \tag{8}$$

where, $W_q \in \mathbb{R}^{(L_m+d_t)\times d_h}$ and $W_{k,\Omega}, W_{v,\Omega} \in \mathbb{R}^{(d_e+d_t)\times d_h}$ are the corresponding weight matrices based on the relation type Ω. The constructed intermediate variables $Q_i(t) \in \mathbb{R}^{1\times d_h}$, $K_{i,\Omega}(t) \in \mathbb{R}^{L_p\times d_h}$ and $V_i(t) \in \mathbb{R}^{(|\mathcal{R}_i(t)|L_p)\times d_h}$ correspond to “Queries”, “Keys”, and “Values” of scaled dot-product attention [36] respectively. It is worth noting that $Q_i(t)$, $K_{i,\Omega}(t)$, and $V_i(t)$ respectively indicate the local information of the resource requirement of terminal i, the neighbor information based on relationship Ω, and the concatenated neighbor information.

(3) *Information fusion*: To obtain ultimate result of node embedding $h_i(t)$, we have devised a mechanism for the fusion of local information and neighbor information. In order to aggregate the information of neighbor nodes, we inject the heterogeneous graph structure into the attention mechanism. Specifically, we separately calculate the importance of neighbor nodes to the target node based on various relations. To make the coefficients easily comparable among different neighbor nodes, we normalize them over all neighbors of the target node using a softmax function. Finally, an multilayer perceptron (MLP) is used to combine the Memory with the aggregated information. The whole process is shown in Fig. 4.

$$\begin{aligned} h_i(t) &= MLP(\widetilde{h^1_i}(t)\|\widetilde{h^2_i}(t)\| \dots \|\widetilde{h^L_i}(t)\|M_i(t)) \\ \tilde{h}^l_i(t) &= softmax(\mathop{\|}_{\Omega\in\mathcal{R}_i(t)} \frac{Q^l_i(t)K^l_{i,\Omega}(t)^T}{\sqrt{d_h}})V^l_i(t) \end{aligned}, \tag{9}$$

where, L is the number of independent execution attention mechanisms. The estimated result of a single attention mechanism $\tilde{h}^l_i(t)$ is obtained by the weighted sum of $V^l_i(t)$. The weight is given by the dot product of Q-K pair, which indicates the importance of neighbor nodes to target node in terms of resource demand.

After obtaining the estimated value of y_c, the shore station needs to allocate an appropriate number of slots to target terminal to maximize the channel efficiency. As discussed in Section 3.2, when making channel allocation decisions, we need to balance the current channel efficiency gains and the collision of RR messages, and this interplay depends on the uncertain maritime communication environment, such as random demands triggered by maritime activities, dynamic changes in terminal density, and so on. In uncertain stochastic environments, most decision problems can be modeled as MDPs [37] and solved using traditional dynamic programming [38] and reinforcement learning algorithms. However, in the scenario of our study, terminals exhibit heterogeneity and dense distribution, while the communication environment is characterized by dynamism. These factors render the computing and storage costs of the above-mentioned algorithms increasingly prohibitive. DRL algorithms, by introducing neural network models as approximate value functions to replace table-based updates, have evolved into an effective solution [39]. Therefore, we will seek the optimal CAS based on DRL algorithms.

5. DRL based CAS

The construction of MDP is a fundamental prerequisite for the application of DRL algorithms. The channel allocation process can be considered as the process of interaction between shore station and communication environment, which can be described as a MDP. Specifically, at each channel c, with given state $s(c)$, the agent (shore station) selects action $a(c)$, receiving a reward $r(c)$ and the new state of communication environment $s(c+1)$. The channel allocation behavior is determined by a policy, π, which maps the states to actions. In this section, we design the corresponding elements as follows:

(1) *State*: State is a variable that describes the communication environment observed by the shore station. When the slots of channel c will be allocated to the target ship station i, state is defined as $s(c) = \{M_i(t_c), \{P_\Psi(t_c)|\phi_1(e_{ij}(t_c)) = \Psi\}, O(t_c), I(t_c)\}$, where t_c denotes the timestamp corresponding to the current RR message. $O(t_c)$ indicates the intensity of network load, which is defined as the average estimations of recent resource demands $O(t_c) = \frac{1}{L_o}\sum_{j=1}^{L_o} h(t_o^j)$, $t_o^1 < t_o^2 < \cdots < t_o^{L_o} < t_c$. $I(t_c) = [N_s(t_c), N_{t_c}, N_x(t_c)]$ represents the information of RAC, where $N_s(t_c)$, $N_c(t_c)$ and $N_x(t_c)$ respectively represent the average statistics for the number of successful RAC slots, collision RAC slots, and newly added RAC slots in the last L_I DCs.

(2) *Action*: Action can be naturally defined as the number of allocated slots, however, in practical scenarios, x_{total} is variable [35], which results in a variable action space, posing a challenge for DRL algorithms [40]. Therefore, we establish an effective mapping relationship to transform the variable action space into a constant action space. As indicated by (1), it is evident that channel efficiency is inherently linked to the ratio of x_c to x_{total}. Without loss of generality, we define the action as the proportion of allocated slots to the maximum number of slots in a single DC, denoted as $a(c) \in [0,1]$. Compared to directly defining the action as the number of allocated slots, this definition effectively mitigates the dependence of the action space on x_{total}, thus enhancing the generality of the constructed MDP.

(3) *Reward*: Reward is defined as the immediate channel efficiency gain obtained by the shore station after executing action $a(c)$, which can be expressed as $r(c) = f(\lceil a(c)x_{total}\rceil, x'_c)$, where $\lceil\cdot\rceil$ represents the ceiling operator.

The return from a state is defined as the sum of discounted future reward $F_c = \sum_{j=c}^{C+c-1}\gamma^{j-c}r(c)$, where $\gamma \in [0,1]$ is a discount factor determining the priority of short-term rewards. Note that the return depends on the actions chosen, and therefore on the policy π_φ, where φ is the parameter. In order to maximize the expected return from the start state $J(\varphi) = \mathbb{E}_{s_j\sim\rho_{\pi_\varphi}, a_j\sim\pi_\varphi}[F_1]$, where ρ_{π_φ} denotes the state visitation distribution for the policy π_φ, we develop a DRL algorithm based CAS. Given that the constructed action belong to a continuous action space, viable DRL algorithms to consider include the off-policy based deep deterministic policy gradient (DDPG) algorithm [41] family and the on-policy based Proximal Policy Optimization algorithm [42] family. To enhance the utilization of experience samples and ensure training stability, we employ an improved variant of the DDPG algorithm, namely the Twin Delayed DDPG (TD3) algorithm [43], in search of the optimal policy.

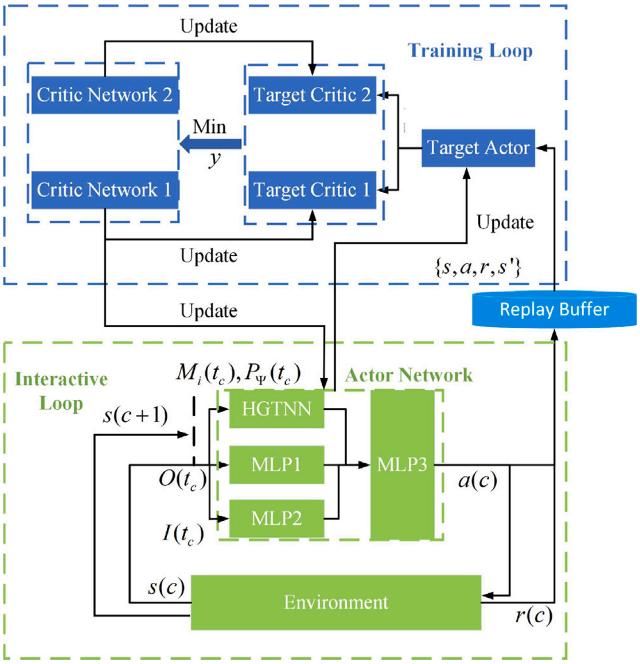


Fig. 5. The pipeline of TD3-based CAS.

TD3 is a well-studied Actor-Critic DRL algorithm including an actor network, two critic networks and their respective target networks. As illustrated in Fig. 5, in the proposed strategy, the actor network is responsible for determining the number of allocated slots, which includes the Communication Environment Estimation Component (CEEC) and the Decision Making Component (DMC). CEEC consists of HTGNN, MLP1 and MLP2. Among them, HTGNN is developed to estimate the y_c of the target terminal, MLP1 is used to capture the short-term trend of traffic load intensity in the current area, and MLP2 is responsible for evaluating the congestion of the current RAC slots. Note that the inputs of HTGNN, MLP1, and MLP2 are $\{M_i(t_c), \{P_\Psi(t_c)|\phi_1(e_{ij}(t_c)) = \Psi\}$, $O(t_c)$, and $I(t_c)\}$. In DMC, MLP3 extracts the outputs of HTGNN, MLP1 and MLP2 to make slot allocation decisions based on comprehensive consideration of current and future channel efficiency gains. Double critic networks are used to measure the performance of current strategy to resolve the problem of overestimation.

The details of training process is described in Algorithm 1. In the initialization stage, we randomly generate the parameters of an actor network π_φ as well as two critic network Q_{θ_1}, Q_{θ_2}, and copy to the respective target networks. In addition, Memory module, PM module and replay buffer $\mathcal{D}$ are initialized with Zero-valued matrix or empty set of the corresponding size. In the training stage, when the shore station allocates the slots of the cth DC, the state $s(c)$ should be constructed according to the proposed HTG model first. Based on the actor network, channel allocation action is obtained as follow:

$$a(c) = \pi_\varphi(s(c)) + \varepsilon_{action}, \tag{10}$$

where $\varepsilon_{action} \sim \mathcal{N}(0,\delta)$ is the exploration noise. After obtaining $r(c)$ and $s(c+1)$, the tuple $\{s(c), a(c), r(c), s(c+1)\}$ is stored as a transition in $\mathcal{D}$. Mini-batch of N_{batch} transitions are randomly sampled to train the network. In TD3, the policy, known as the actor network, can be updated through the deterministic policy gradient (DPG) [44] algorithm:

$$\nabla_\varphi J(\varphi) = \mathbb{E}_{s\sim\rho_{\pi_\varphi}}[\nabla_a Q^{\pi_\varphi}(s,a)|_{a=\pi_\varphi(s)}\nabla_\varphi\pi_\varphi(s)], \tag{11}$$

Algorithm 1 Training Process of the Proposed Framework

```
Initialization
Initialize the critic networks $Q_{\theta_1}$, $Q_{\theta_2}$, and actor network $\pi_\varphi$ with random parameters$\theta_1$, $\theta_2$, $\varphi$
Initialize target networks $\theta'_1 \leftarrow \theta_1$, $\theta'_2 \leftarrow \theta_2$, $\varphi' \leftarrow \varphi$
Initialize PM module, Memory module, replay buffer$\mathcal{D}$
Training
for RR message number $c = 1 : C$ do
    Obverse $s(c)$ and select action $a(c)$ according to (10)
    Obtain reward $r(c)$ and next state $s(c+1)$
    Store tuple $\{s(c), a(c), r(c), s(c+1)\}$ in $\mathcal{D}$
    if $|\mathcal{D}| > N_{batch}$ then
        Sample mini-batch of $N_{batch}$ $\{s, a, r, s'\}$from $\mathcal{D}$
        Calculate the target value of $(s, a)$ according to (13)
        Update critic networks $\theta_i \leftarrow min_{Q_i} \frac{1}{N_{batch}} \sum (y - Q_{\theta_i}(s,a))^2$
        if $n$ mod $d = 0$ then
            Update $\varphi$ by DPG according to (14)
            Update target networks $\theta'_i \leftarrow \mu\theta_i + (1-\mu)\theta'_i$, $\varphi' \leftarrow \mu\varphi + (1-\mu)\varphi'$
        end if
    end if
end for
```

where $Q^{\pi_\varphi}(s,a) = \mathbb{E}_{s_j \sim \rho_{\pi_\varphi}, a_j \sim \pi_\varphi}[F_c | s, a]$ represents the Q value that is defined as the expected return when performing action a in state s and following policy π_φ. Q-learning [45] is applied to train the Q value function approximator, known as the critic networks. The parameters of critic networks are optimized by minimizing the loss:

$$L(\theta_i) = (y - Q_{\theta_i}(s,a))^2, i = 1, 2 \tag{12}$$

where y is the target Q value. Considering that similar actions should have similar values, TD3 adds clipped noise to the target action to reduce estimation error. Therefore, the target Q value of selected action is defined as:

$$y = r + \gamma min_{i=1,2} Q_{\theta'_i}(s', a'), \tag{13}$$

where $a' = \pi_{\varphi'}(s') + \epsilon_{policy}$, $\epsilon_{policy} \sim clip(\mathcal{N}(0, \delta'), -u, u)$. Consequently, the DPG of policy π_φ can be reformulated as:

$$\nabla_\varphi J(\varphi) \approx \frac{1}{N_{batch}} \sum \nabla_a Q_{\theta_i}(s,a)|_{a=\pi_\varphi(s)} \nabla_\varphi \pi_\varphi(s). \tag{14}$$

In addition, delayed policy updates and soft target updates are employed to further reduce the variance of estimation error.

6. Simulation

In this section, we demonstrate the performance of the proposed algorithm in the shore station communication scenario of VDE-TER based on real Automatic Identification System (AIS) data, all experiments are trained and evaluated on a full fledged computer (CPU: Intel(R) Core(TM) i7-12700H, GPU: NVIDIA GeForce RTX 3080Ti).

6.1. Datasets

As the VDES system is still under development, as far as we know, publicly available traffic datasets for the VDE-TER service are currently non-existent. Therefore, to perform simulations, we generate the necessary datasets based on the AIS ship trajectory data.

We establish a traffic model of VDE-TER service based on existing international standards, which is then used to generate the traffic data for each active ship station. The potential service types of VDE-TER [33] and their characteristics are summarized in Table 1. Given the absence of a clear definition of service priority for VDE-TER within current relevant standards, for the sake of simplicity, we have divided those services into two priority levels based on the relevance of the service content and navigational safety.

Table 1
Service feature table.

Service type	Packet size	Priority
Search and Rescue communications	2,14	1
Safe related information	8	1
Ship reporting	8	1
Vessel traffic services	2,8	1
Route exchange	2	1
Chart and Publication	Pareto distribution	2
Logistics/Services	Pareto distribution	2

Table 2
Filter parameters.

Time	Longitude (°)	Latitude (°)
March 15, 2021	121.73∼122.42	30.33∼30.92

(1) *Priority level 1*: Priority level 1 refers to the basic services which are essential for safe navigation. All stations should be able to transmit those services. To ensure data standardization, the data packets for such services should remain as consistent as possible. Generally, the transmitted content is generated based on the pre-defined data format. We set the packet size of these services according to their application scenarios, wherein 2(slots), 4(slots), and 8(slots) correspond to small, medium and large data packets respectively.

(2) *Priority level 2*: Priority level 2 refers to proprietary services provided according to user needs, for which the data packet size should dynamically adjust based on the service content. Referring to the "Traffic models for Cellular IoT" section in [46], we assume that the packet size of those services obeys the Pareto distribution:

$$P(X > x) = (\frac{x}{x_{min}})^{-k}, \tag{15}$$

where k and x_{min} denote the shape parameter and the minimum size of packet, respectively. It is expected that the service demand will vary with ship station type, e.g. passenger ships have much lower demand for logistics services than cargo ships. We randomly generate the minimum packet size of each proprietary services for different ship type, that is, $x^k_{min}(i) \sim U(0, x_{total})$, where k and i denote the service type and ship type, respectively.

The arrival rate of VDE-TER services is expected to vary depending on the status of the ship station. For example, the demand of Route Exchange service will be much higher in the sailing state than that in the anchoring state. A similar provision is presented in [47]. As with most related work [48], we assuming that the arrival of each services obeys the Poisson distribution. Specifically, we randomly select $\lambda^k(j)$ from $\{0.001, 0.01, 0.1, 1.0\}$, where j denote the status of ships.

In order to obtain relevant information about active ship stations in the area covered by shore stations, we filter the AIS ship trajectory data. With Yangkou Port as the center and a square with a side length of 50 nautical miles as the effective coverage area, we screened the ship trajectory from 00:00:00 to 23:59:59 on March 15, 2021. The relevant parameters are shown in Table 2. Based on the established traffic model, we select ShipType and Status as the features of ship stations, some examples of AIS data are shown in Table 3. It is worth noting that if the AIS information of a ship station is not received for more than 2 h, it is considered that the ship station is not activated or has left the current area. We divide the datasets into 24 groups by hour, the data of first 30 min of each group used for training while the last 30 min are used for evaluating. As shown in Fig. 6, we counted the number of new resource requests per hour in this area, which indicates the traffic load intensity of each data group.

Table 3
Part of real samples.

MMSI	TimeStamp	Longitude (°)	Latitude (°)	ShipType	Status
412407490	2021/3/15 0:00:01	122.27678	30.43727	Passenger ship	Under way using engine
413225490	2021/3/15 0:00:17	122.01147	30.65538	Fishing vessel	Moored
412431550	2021/3/15 0:01:02	121.88949	30.38414	Tanker	Under way using engine
412047670	2021/3/15 13:57:01	122.08180	30.60821	Towing vessel	Under way using engine
413447450	2021/3/15 13:57:06	122.40177	30.64634	Cargo ship	At anchor

Table 4
Service feature table.

Parameter	Value	Description
x_{total}	14	Total number of slots in a channel
d_h	128	Dimensions of intermediate variables
d_t	32	Dimensions of time encoding
d_e	64	Dimensions of $h_i(t)$
L_m	10	Length of Memory
L_p	10	Length of Public Mailbox
L_o	10	Length of $O(t_c)$ statistics window
L_I	3	Length of $I(t_c)$ statistics window
L	2	Number of multi-heads
α	2	Gain factor of DC
β	0.4	Gain factor of RAC
γ	0.98	Discount factor
δ	0.4	Variance of action noise
δ'	0.4	Variance of policy noise
u	0.5	Range of action noise
d	4	Delay update factor
lr	1e−4	Learning rate
n_{batch}	256	Size of mini-batch
μ	0.005	Update step size of target networks

Table 5
Time consumption.

HTG	HGT	TGAT	LSTM
1.37 ms	1.35 ms	1.30 ms	1.12 ms

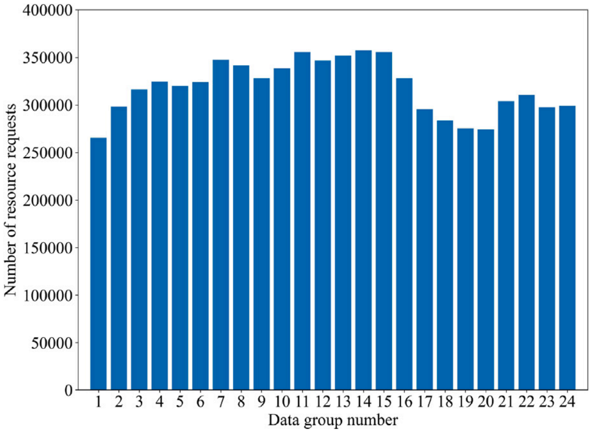


Fig. 6. Statistical results of new resource requests.

6.2. Simulation settings

According to [35], we adopt the default slot functions and the frame structure is set as follows: each minute has 25 frames, each frame has 5 channels and 12 fixed RAC slots.

The structure of the actor network is set as follow:

(1) *MLP1*: MLP1 consists of 2 fully-connected (FC) layers, whose structure is $\{d_h \times 2d_h, 2d_h \times d_e\}$.

(2) *MLP2*: MLP2 consists of 2 fully-connected (FC) layers, whose structure is $\{3 \times d_h, d_h \times d_e\}$.

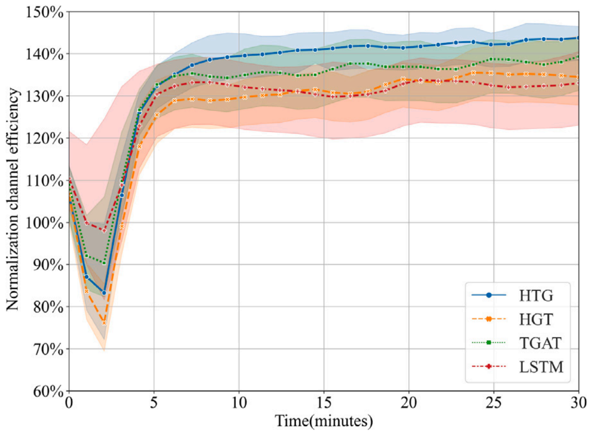


Fig. 7. Normalization channel efficiency.

(3) *MLP3*: MLP3 consists of 3 FC layers, whose structure is $\{3d_e \times 2d_h, 2d_h \times d_h, d_h \times 1\}$ (The critic network modifies the first size to $(3d_e + 1)$).

In addition, the MLP of HTGNN is composed of 3 FC layers, whose structure is $\{(Ld_h + L_m) \times 2d_h, 2d_h \times d_h, d_h \times d_e\}$, and the activation function between FC layers is ReLU. Other simulation parameters are summarized in Table 4.

We compare the proposed strategy with 4 other strategies as follow:

(1) *FA*: Fixed allocation (FA) Strategy is the default channel allocation strategy in existing standards [35], which allocates the maximum number of time slots to all received requests.

(2) *LSTM*: Based on the proposed channel allocation strategy, we replace the HTGNN with LSTM [49], and Memory is regarded as time series data as the input of LSTM correspondingly. The number of features in the hidden state is set to d_h.

(3) *HGT*: Similarly, we use heterogeneous graph transformer (HGT) [18] instead of HTGNN. Compared with HTGNN, HGT lacks consideration of the time factor.

(4) *TGAT*: Temporal graph attention layer (TGAT) [21] is adopted to replace the HTGNN, which does not distinguish neighbor nodes with different relations.

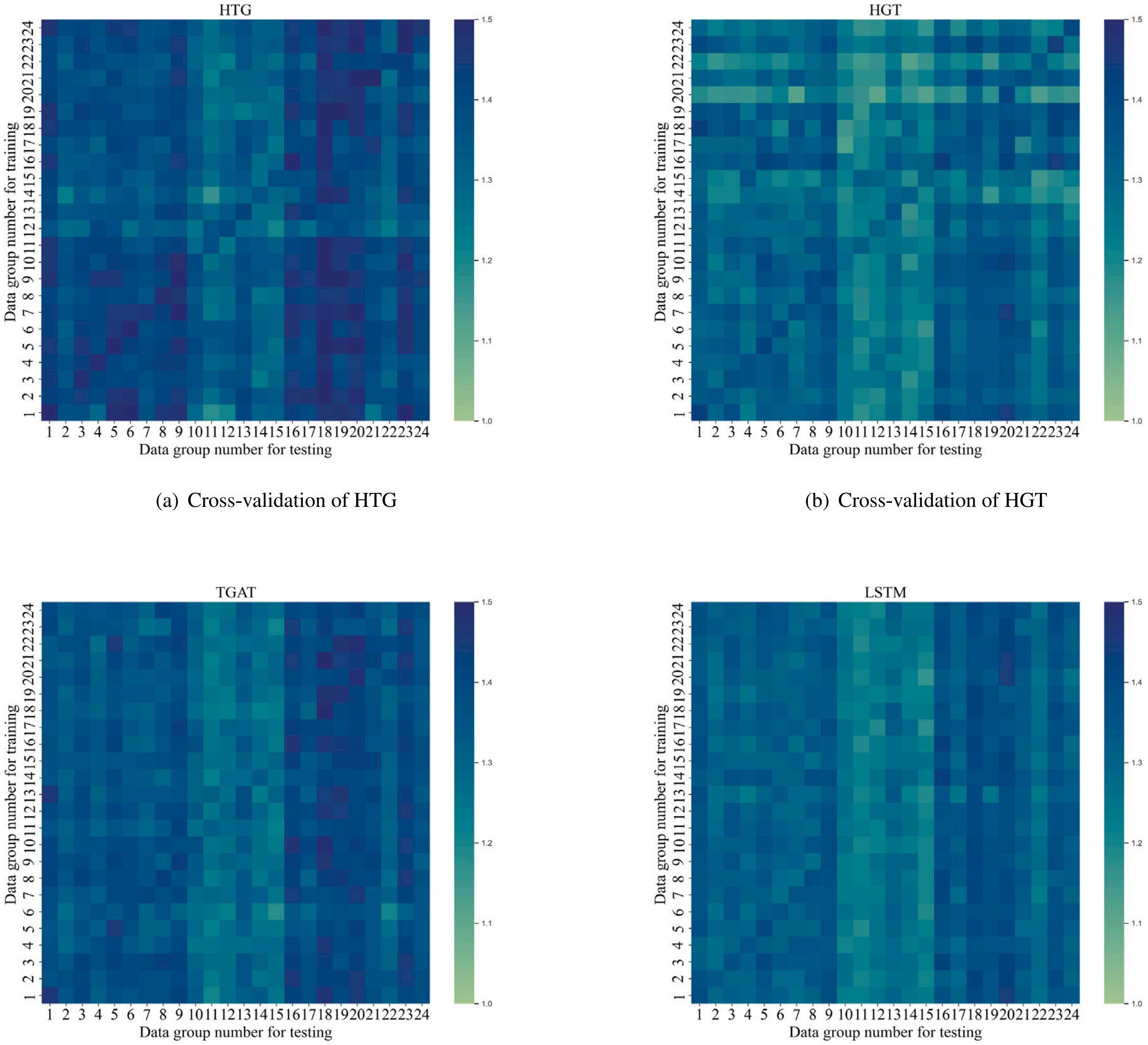
6.3. Simulation results

To fully demonstrate the performance of the proposed strategy, we define 4 evaluation indicators.

(1) *Normalization channel efficiency*: It can be seen from (2) that the channel efficiency is related to the total resources demand in corresponding time period. In order to avoid the impact of dynamic traffic on performance evolution, we normalize the total channel efficiency of each strategy over different time periods, which can be expressed as:

$$\widetilde{U}_{\mathcal{K}} = \frac{1}{J} \sum_{j=1}^{J} \frac{U_{\mathcal{K}}^{j}}{U_{FA}^{j}}, \tag{16}$$

where $\mathcal{K}$ and $J = 24$ denote the evaluated strategy and the total number of data groups, respectively. It is worth noting that the normalization channel efficiency (NCE) measures the improvement of the current strategy compared to the FA strategy.



(a) Cross-validation of HTG

(b) Cross-validation of HGT

(c) Cross-validation of TGAT

(d) Cross-validation of LSTM

Fig. 8. Cross-validation of different strategies.

(2) *RMSE*: Root Mean Squared Error (RMSE) is employed to measure the deviation between x_c and y_c, which indicates the performance of CAS in terms of the current channel efficiency.

$$RMSE = \sqrt{\frac{1}{C}\sum_{c=1}^{C}\left(y_c - x_c\right)^2} \tag{17}$$

(3) *CPRA*: Collision probability of RAC slots (CPRA) is defined as the ratio of the number of RAC slots collided to the total number of RAC slots, which measures the ability of the CAS to mitigate the congestion problem of RAC slots.

(4) *Time consumption*: In the actual communication scenario, since the shore station needs to allocate slots to the target ship station in real time, the time consumption of making decisions is an essential factor in evaluating the feasibility of the strategy. We define the time consumption as the average inference time of the actions chosen.

Fig. 7 compares the NCE of different strategies. We conducted the experiments on 24 data groups separately, and averaged the results of each strategy. From the presented simulation results, we can see that our proposed HTGNN-powered CAS (called HTG) obtains the highest NCE compared to TGAT, HGT and LSTM. Moreover, the convergence of HTG superior to other strategies. In fact, as the training progresses, that is, the actual scene evolves over time, the time-vary ship status will dynamically change the resource demand preference of the ship station, while the mobility of the ship station will dynamically change the number of ship stations and the distribution of ship types. On the one hand, owing to the application of the time encoding technology,

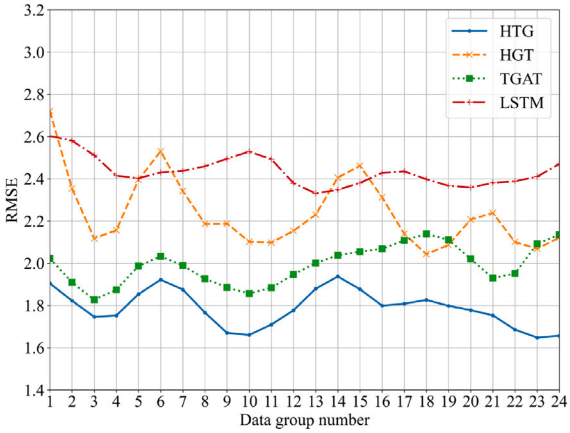


Fig. 9. CPRA of different data groups.

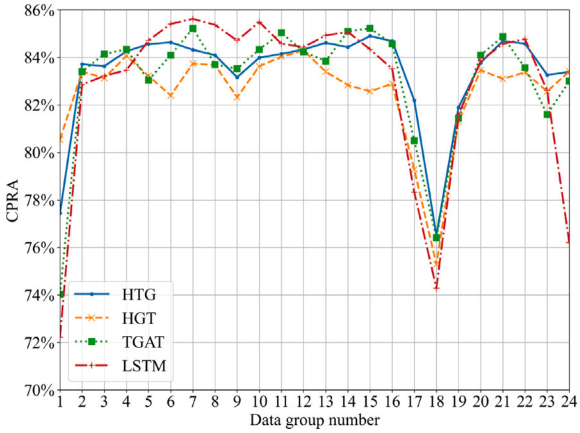


Fig. 10. CPRA of different data groups.

HTG possesses a strong adaptive capability to the data group whose communication environment changes significantly with time by taking into account the temporal information. On the other hand, due to the fact that HTG aggregates neighbor information base on heterogeneous relationships, it can distinguish terminal heterogeneity, i.e., resource requirements vary with ship types. However, HGT, TGAT and LSTM either does not consider the temporal information of the network or does not consider the heterogeneous topology information of the network. The trained parameters of these models cannot be adapted to new sample distributions, limiting the performance of the strategy.

To further evaluate the generalization ability of strategies between data groups, we adopt a cross-validation method, that is, train on one data group and test on all other data groups. As shown in Fig. 8, we plot the results of the cross-validation as a heatmap. Among them, the ith row represents the NCE of the policy trained based on the ith data group on different data groups, and the jth column represents the NCE of the policies trained based on different data groups on the jth data group. According to the simulation results, HTG has higher thermal values on off-diagonal elements than other strategies, which indicates that HTG can obtain higher NCE on data groups that are not visible during training, proving that HTG has better performance in terms of generalization ability.

In addition, in order to investigate the direct reasons for the superior performance of the proposed strategy, according to (2), we compare the performance of different strategies from the perspective of RMSE and CPRA. As shown in Figs. 9 and 10, it can be seen from the results that HTG obtains significantly lower RMSE at a very small CPRA cost, which means HTG occupies few potential resources of the RAC slot while satisfying the current channel efficiency as much as possible.

Finally, we compare the real-time inference time consumption of different strategies. As shown in Table 5, HTG has the largest time consumption, 1.5%, 5.4% and 22.3% more than HGT, TGAT, and LSTM respectively, Nevertheless, 1.37 ms is still an absolutely small processing delay. As far as the VDE-TER scenario considered in this paper is concerned, it meets the processing delay requirements (no more than 5 slots, e.t., 5×26.67 ms $= 133.33$ ms). This demonstrates the feasibility of the proposed strategy in a real-world maritime IoT system.

7. Conclusion

Due to the heterogeneous network and dynamic environment, designing an intelligent CAS for the maritime IoT system is an extremely tough problem. In this paper, we propose a HTGNN-model-driven DRL algorithm tailored for flexible channel allocation, aiming to obtain long-term maximization of channel efficiency. Since the heterogeneous and dynamic communication scenario are modeled as HTG, the proposed HTGNN model can effectively investigate the temporal and topological information of the network to estimate the resource demands of target terminals. Considering the random evolution of communication environment and the inherent correlation among channel allocation behaviors, we learn the CAS with the maximum channel efficiency returns based on the DRL algorithm by interacting with the environment. To verify the performance of the proposed strategy, we conducted simulation experiments grounded in a real-world maritime IoT scenario. Simulation results show that the proposed strategy can effectively improve the channel utilization compared with the baseline strategy. In addition, compared with other state-of-the-art neural network models, the established HTGNN model displays a more robust node embedding representation and superior generalization capacity.

CRediT authorship contribution statement

Zongwang Li: Conceptualization, Methodology, Experimentation, Writing – original draft, Simulation analysis. **Zhuochen Xie:** Conceptualization, Simulation analysis. **Xiaohe He:** Experimentation, Writing – review & editing. **Xuwen Liang:** Supervision, Methodology, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB2902603 and in part by the Science and Technology Innovation Plan of Shanghai Science and Technology Commission under Grant 22511100500.

References

[1] T. Xia, M.M. Wang, J. Zhang, L. Wang, Maritime internet of things: Challenges and solutions, IEEE Wirel. Commun. 27 (2) (2020) 188–196, http://dx.doi.org/10.1109/MWC.001.1900322.

[2] IMO, MSC 85/26/Add.1 Annex 20, Strategy for the Development and Implementation of E-Navigation, 2008.

[3] M.M. Wang, J. Zhang, X. You, Machine-type communication for maritime internet of things: A design, IEEE Commun. Surv. Tutor. 22 (4) (2020) 2550–2585, http://dx.doi.org/10.1109/COMST.2020.3015694.

[4] T. Yang, J. Chen, N. Zhang, AI-empowered maritime internet of things: A parallel-network-driven approach, IEEE Netw. 34 (5) (2020) 54–59.

[5] J. Suarez-Varela, P. Almasan, M. Ferriol-Galmes, K. Rusek, F. Geyer, X. Cheng, X. Shi, S. Xiao, F. Scarselli, A. Cabellos-Aparicio, P. Barlet-Ros, Graph neural networks for communication networks: Context, use cases and opportunities, IEEE Netw. (2022) 1–8, http://dx.doi.org/10.1109/MNET.123.2100773.

[6] R.W. Liu, J. Nie, S. Garg, Z. Xiong, Y. Zhang, M.S. Hossain, Data-driven trajectory quality improvement for promoting intelligent vessel traffic services in 6G-enabled maritime IoT systems, IEEE Internet Things J. 8 (7) (2021) 5374–5385, http://dx.doi.org/10.1109/JIOT.2020.3028743.

[7] R.W. Liu, M. Liang, J. Nie, W.Y.B. Lim, Y. Zhang, M. Guizani, Deep learning-powered vessel trajectory prediction for improving smart traffic services in maritime internet of things, IEEE Trans. Netw. Sci. Eng. 9 (5) (2022) 3080–3094, http://dx.doi.org/10.1109/TNSE.2022.3140529.

[8] M.M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, Geometric deep learning: Going beyond euclidean data, IEEE Signal Process. Mag. 34 (4) (2017) 18–42, http://dx.doi.org/10.1109/MSP.2017.2693418.

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (1) (2021) 4–24, http://dx.doi.org/10.1109/TNNLS.2020.2978386.

[10] W. Jiang, Graph-based deep learning for communication networks: A survey, Comput. Commun. 185 (2022) 40–54, http://dx.doi.org/10.1016/j.comcom.2021.12.015, URL https://www.sciencedirect.com/science/article/pii/S0140366421004874.

[11] Y. Shen, J. Zhang, K.B. Letaief, How neural architectures affect deep learning for communication networks? in: ICC 2022 - IEEE International Conference on Communications, 2022, pp. 389–394, http://dx.doi.org/10.1109/ICC45855.2022.9839205.

[12] J. Li, P. Sun, Y. Hu, Traffic modeling and optimization in datacenters with graph neural network, Comput. Netw. 181 (2020) 107528, http://dx.doi.org/10.1016/j.comnet.2020.107528, URL https://www.sciencedirect.com/science/article/pii/S1389128620311865.

[13] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations (ICLR), 2017.

[14] W.L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS '17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 1025–1035.

[15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018, URL https://openreview.net/forum?id=rJXMpikCZ.

[16] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P.S. Yu, Heterogeneous graph attention network, in: The World Wide Web Conference, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 2022–2032, http://dx.doi.org/10.1145/3308558.3313562.

[17] C. Zhang, D. Song, C. Huang, A. Swami, N.V. Chawla, Heterogeneous graph neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Amp; Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 793–803, http://dx.doi.org/10.1145/3292500.3330961.

[18] Z. Hu, Y. Dong, K. Wang, Y. Sun, Heterogeneous graph transformer, in: Proceedings of the Web Conference 2020, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 2704–2710.

[19] A. Zaki, M. Attia, D. Hegazy, S. Amin, Comprehensive survey on dynamic graph models, Int. J. Adv. Comput. Sci. Appl. 7 (2) (2016) http://dx.doi.org/10.14569/IJACSA.2016.070273.

[20] Y. Zhu, F. Lyu, C. Hu, X. Chen, X. Liu, Encoder-decoder architecture for supervised dynamic graph learning: A survey, 2022, arXiv:2203.10480.

[21] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, K. Achan, Inductive representation learning on temporal graphs, in: International Conference on Learning Representations (ICLR), 2020.

[22] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, M. Bronstein, Temporal graph networks for deep learning on dynamic graphs, in: ICML 2020 Workshop on Graph Representation Learning, 2020.

[23] X. Wang, D. Lyu, M. Li, Y. Xia, Q. Yang, X. Wang, X. Wang, P. Cui, Y. Yang, B. Sun, Z. Guo, APAN: Asynchronous propagation attention network for real-time temporal graph embedding, in: Proceedings of the 2021 International Conference on Management of Data, SIGMOD '21, 2021, pp. 2628–2638.

[24] Q. Li, Y. Shang, X. Qiao, W. Dai, Heterogeneous dynamic graph attention network, in: 2020 IEEE International Conference on Knowledge Graph (ICKG), 2020, pp. 404–411, http://dx.doi.org/10.1109/ICBK50248.2020.00064.

[25] Y. Fan, M. Ju, C. Zhang, Y. Ye, Heterogeneous temporal graph neural network, in: Proceedings of the 2022 SIAM International Conference on Data Mining (SDM), 2022, pp. 657–665.

[26] T. Chen, X. Zhang, M. You, G. Zheng, S. Lambotharan, A GNN-based supervised learning framework for resource allocation in wireless IoT networks, IEEE Internet Things J. 9 (3) (2022) 1712–1724, http://dx.doi.org/10.1109/JIOT.2021.3091551.

[27] J. Guo, C. Yang, Learning power allocation for multi-cell-multi-user systems with heterogeneous graph neural networks, IEEE Trans. Wireless Commun. 21 (2) (2022) 884–897, http://dx.doi.org/10.1109/TWC.2021.3100133.

[28] X. Zhang, Z. Zhang, L. Yang, Learning-based resource allocation in heterogeneous ultradense network, IEEE Internet Things J. 9 (20) (2022) 20229–20242, http://dx.doi.org/10.1109/JIOT.2022.3173210.

[29] M.S. Frikha, S.M. Gammar, A. Lahmadi, L. Andrey, Reinforcement and deep reinforcement learning for wireless internet of things: A survey, Comput. Commun. 178 (2021) 98–113, http://dx.doi.org/10.1016/j.comcom.2021.07.014, URL https://www.sciencedirect.com/science/article/pii/S0140366421002681.

[30] K. Nakashima, S. Kamiya, K. Ohtsu, K. Yamamoto, T. Nishio, M. Morikura, Deep reinforcement learning-based channel allocation for wireless lans with graph convolutional networks, IEEE Access 8 (2020) 31823–31834, http://dx.doi.org/10.1109/ACCESS.2020.2973140.

[31] U. Gunarathna, R. Borovica-Gajic, S. Karunasekara, E. Tanin, Solving Dynamic Graph Problems with Multi-Attention Deep Reinforcement Learning, 2022, arXiv e-prints arXiv:2201.04895.

[32] S. Munikoti, D. Agarwal, L. Das, M. Halappanavar, B. Natarajan, Challenges and opportunities in deep reinforcement learning with graph neural networks: A comprehensive review of algorithms and applications, 2022, arXiv:2206.07922.

[33] IALA, IALA Guideline G1117 Edition 3.0, VHF Data Exchange System (VDES) Overview, 2022.

[34] F. Lazaro, R. Raulefs, W. Wang, F. Clazzer, S. Plass, VHF data exchange system (VDES): An enabling technology for maritime communications, CEAS Space J. 11 (2019) 55–63, http://dx.doi.org/10.1007/s12567-018-0214-8.

[35] ITU, Document ITU-R M.2092-1, Technical Characteristics for a VHF Data Exchange System in the VHF Maritime Mobile Band, 2022.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS '17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6000–6010.

[37] M.L. Puterman, Markov decision processes, Handb. Oper. Res. Manag. Sci. 2 (1990) 331–434.

[38] D. Bertsekas, Dynamic Programming and Optimal Control: Volume I, vol. 4, Athena scientific, 2012.

[39] N.C. Luong, D.T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, D.I. Kim, Applications of deep reinforcement learning in communications and networking: A survey, IEEE Commun. Surv. Tutor. 21 (4) (2019) 3133–3174, http://dx.doi.org/10.1109/COMST.2019.2916583.

[40] S. Huang, S. Ontañón, A closer look at invalid action masking in policy gradient algorithms, 2020, arXiv preprint arXiv:2006.14171.

[41] T. Lilicrap, J. Hunt, A. Pritzel, N. Hess, T. Erez, D. Silver, Y. Tassa, D. Wiestra, Continuous control with deep reinforcement learning, in: International Conference on Representation Learning (ICRL), 2016.

[42] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, 2017, arXiv preprint arXiv:1707.06347.

[43] S. Fujimoto, H. Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: International Conference on Machine Learning, PMLR, 2018, pp. 1587–1596.

[44] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, M. Riedmiller, Deterministic policy gradient algorithms, in: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML '14, JMLR.org, 2014, pp. I–387–I–395.

[45] C.J. Watkins, P. Dayan, Q-learning, Mach. Learn. 8 (1992) 279–292.

[46] 3GPP, 45.820 V13.1.0, Cellular System Support for Ultra-Low Complexity and Low Throughput Internet of Things (CIoT), 2015.

[47] ITU, Document ITU-R M.1371-5, Technical Characteristics for an Automatic Identification System using Time Division Multiple Access in the VHF Maritime Mobile Frequency Band, 2014.

[48] P. Fazio, M. Mehic, M. Voznak, F. De Rango, M. Tropea, A novel predictive approach for mobility activeness in mobile wireless networks, Comput. Netw. 226 (2023) 109689, http://dx.doi.org/10.1016/j.comnet.2023.109689, URL https://www.sciencedirect.com/science/article/pii/S1389128623001342.

[49] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.