Wireless Networks

Liang Xiao
Helin Yang
Weihua Zhuang
Minghui Min

Reinforcement Learning for Maritime Communications

Springer

Wireless Networks

Series Editor

Xuemin Sherman Shen, University of Waterloo, Waterloo, ON, Canada

The purpose of Springer's Wireless Networks book series is to establish the state of the art and set the course for future research and development in wireless communication networks. The scope of this series includes not only all aspects of wireless networks (including cellular networks, WiFi, sensor networks, and vehicular networks), but related areas such as cloud computing and big data. The series serves as a central source of references for wireless networks research and development. It aims to publish thorough and cohesive overviews on specific topics in wireless networks, as well as works that are larger in scope than survey articles and that contain more detailed background information. The series also provides coverage of advanced and timely topics worthy of monographs, contributed volumes, textbooks and handbooks.

Liang Xiao • Helin Yang • Weihua Zhuang •
Minghui Min

Reinforcement Learning for Maritime Communications

Springer

Liang Xiao
Department of Information
and Communication Engineering
Xiamen University
Xiamen, Fujian, China

Helin Yang
Department of Information
and Communication Engineering
Xiamen University
Xiamen, Fujian, China

Weihua Zhuang
Wireless Communication Networks
University of Waterloo
Waterloo, ON, Canada

Minghui Min
School of Information and Control
Engineering
China University of Mining and Technology
Xuzhou, Jiangsu, China

ISSN 2366-1186 ISSN 2366-1445 (electronic)
Wireless Networks
ISBN 978-3-031-32137-5 ISBN 978-3-031-32138-2 (eBook)
https://doi.org/10.1007/978-3-031-32138-2

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book provides a broad coverage of maritime wireless communication issues, such as reliability, security, resource management, and privacy protection. Reinforcement learning enables maritime communication systems to address these issues. This book includes four chapters from international researchers working in this area. Professionals and researchers can find *Reinforcement Learning for Maritime Communications* a useful reference. The material serves as a useful reference for researchers, graduate students, and practitioners seeking solutions to maritime wireless communication and information security-related issues.

In Chap. 1, we mainly introduce the development, related work, and challenges of maritime wireless communications. The motivation, objective, and major contributions of this book are provided.

In Chap. 2, we present the system model and problem formulation of intelligent reflecting surface aided maritime wireless communications. A reinforcement learning-based solution is provided and its performance is evaluated via theoretic analysis and simulation results.

In Chap. 3, we review the related work and challenges of privacy protection for the maritime Internet of Things (IoT) offloading process and propose a reinforcement learning-based privacy-aware offloading scheme to help IoT devices protect both the user location and the service usage pattern privacy.

In Chap. 4, we describe a reinforcement learning-based resource management for ultra-reliable low-latency maritime communications. The resource management algorithm design principle and simulation results are analyzed as well.

In Chap. 5, we review the related work and challenges to protect the semantic location privacy for the location-based services and propose a reinforcement learning-based location privacy protection scheme. Differential privacy is applied to randomize the released maritime devices' locations and the perturbation policy is optimized to improve both the privacy protection level and the service performance.

In Chap. 6, we draw conclusions of this book and identify the future research directions.

This book investigates various research topics, including intelligent reflecting surface-aided communications, privacy-aware IoT communications, intelligent resource management, and semantic location perturbation for maritime communications.

This book could not have been made possible without the contributions from Shiyu Xu, Shi Yu, Siyao Li, and Jingchen Xu in the book writing and valuable discussions. We would also like to thank all the colleagues whose work enlightened our thoughts and research presented in this book. Arun Siva Shanmugam.

Xiamen, Fujian, China Liang Xiao
Xiamen, Fujian, China Helin Yang
Waterloo, ON, Canada Weihua Zhuang
Xuzhou, Jiangsu, China Minghui Min
October 2022

Contents

Chapter 1
Introduction

Maritime communication systems have attracted ever-increasing research attention and become an important part of the fifth-/sixth-generation (5G/6G) communications. Maritime communication networks support ship-to-ship, ship-to-shore, and ship-to-sensor communications, where channel interference mitigation, reliable communication, secure transmission, and jamming resistance are important. In this chapter, we review related works and summarize the main content of this book.

1.1 Maritime Communications

The development of maritime communication systems depends on the reliable and secure communication under various quality of service (QoS) requirements of the booming maritime services, such as smart maritime surveillance, auto navigations, maritime leisure services, and electronic chart managements.

With the increasing number of ships, vessels, offshore platforms, and Internet of Things (IoTs) equipment, the demand for high-speed and ultrareliable maritime communication is growing under large-scale dynamic maritime networks. A large amount of data are produced by maritime applications, such as the surveillance videos collected from bridge, engine room, or other critical regions of a vessel. For instance, text, voice, and video data have to be transmitted in real time with ultralow bit error rates to support the ship-to-ship and ship-to-shore coordination.

1.1.1 Related Works

Marine communications, including the ship-to-ship, ship-to-shore, ship-to-satellite, and ship-to-sensor communications, are required to support services with various

L. Xiao et al., *Reinforcement Learning for Maritime Communications*, Wireless Networks, https://doi.org/10.1007/978-3-031-32138-2_1

QoS requirements such as in terms of the data rates, packet loss rates, bit error rates (BER), energy consumption, and spectral efficiency. For example, a navigational automated identification system applies the high-frequency (HF)-, very-high-frequency (VHF)-, and ultrahigh-frequency (UHF)-based shore-to-ship communications to support and provide voice, fax, and text services for ships up to 740 kilometers (km) away from shores [1–3]. The long-term evolution (LTE) system applies orthogonal frequency-division multiplexing (OFDM) and carrier aggregation to provide over 10 Mbps broadband multimedia services[4]. The Windsurfing World Cup based on 5G systems operating at 28 GHz relays the 4K resolution video stream from ships to ground base stations [5].

Vessel networks consist of vessels, beacons, and buoys that expand the communication coverage and reliability against severe channel attenuation due to ultra-long communication distances and obstacles. For instance, mobile ad hoc networks (MANETs) support up to five hops between 27 MHz and 40 MHz to carry low-speed data at sea off Nishinomiya and Kushimoto [6]. A satellite-aided mesh network among the adjacent vessels, maritime beacons, and buoys uses IEEE 802.16 to provide broadband services for up to 225 km^2 maritime area [7].

Satellite communication systems establish the communication link between the far-offshore vessels and the satellites in low Earth orbit (LEO), medium Earth orbit (MEO), geostationary orbit (GEO), and high elliptical orbit (HEO) [8]. As an important GEO-based communication system, Inmarsat-6 provides both 2 GHz and 28 GHz dual service with data rates up to 60 Mbps [9]. The first MEO system, O3B, operates on 20 satellites, at 28 GHz, to provide 500 Mb broadband maritime data services with less than 140 ms latency [10]. Consisting of 66 LEO satellites, iridium, provides reliable and low-latency maritime safety communications for global coverage [11].

Due to the deployment flexibility and the line-of-sight (LoS) propagation, unmanned aerial vehicles (UAVs) and balloons act as relays or aerial base stations (BSs) to enhance the maritime communication coverage, data rates, and reliability [12]. As an example of multi-hop air-based maritime communications, BLUECOM+ uses tethered balloons to provide broadband communication services [13]. The UAV trajectory control based on reinforcement learning (RL) improves the maritime communication performance to support emergency field and radar surveillance under complicated maritime environments [14–16].

In a general maritime wireless communication system model as shown in Fig. 1.1, the ships choose the UAVs as relay to connect to the BSs at an island and use the satellite communications as the backup. The space-air-ground hybrid network consists of the shore-based maritime communication system, satellite-based system, and air-based system to enhance the coverage and user capacity [17]. For instance, the tower-borne cellular BSs and the space-air-ground hybrid network provide wide-area seamless coverage for offshore maritime devices [18]. The data rate, energy efficiency, and coverage of the 6G network can be improved by a joint link scheduling and rate control algorithm as presented in [19] with the QoS guarantee.

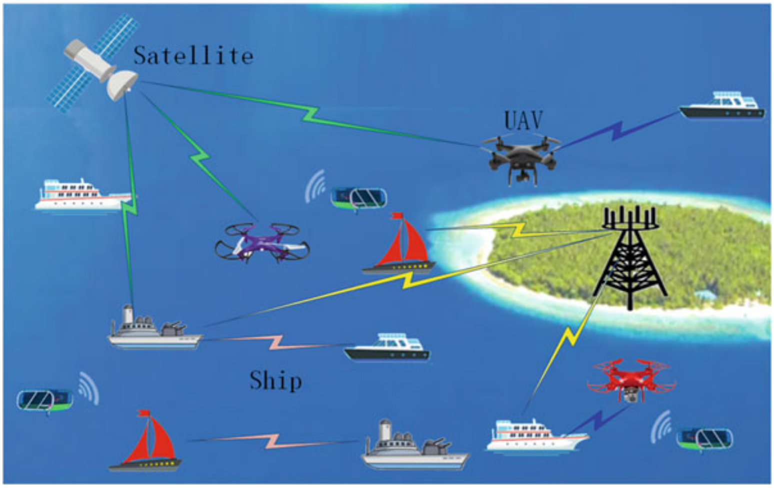


Fig. 1.1 Illustration of maritime wireless communications

Recently, machine learning (ML) has gained huge attention from industry and academia. The development of ML has inspired great research efforts toward the realization of maritime communication services under the heterogeneous maritime network architecture [20]. The accuracy of modulation recognition will severely degrade in multipath fading channels, which is a critical problem for maritime wireless communications. A novel blind equalization-aided deep learning (DL) proposed in [21] employs the structure of ResNet and improves the recognition accuracy under severe multipath scenarios.

RL is a type of ML which can achieve the optimal strategy via trial and errors without the knowledge of the environment. This characteristic is suitable for the complex marine environment and results in more extensive attention as compared with other methods. With the rapid enhancement of maritime communication services, IoT devices are improving synchronously, and QoS of data transmission becomes a bottleneck, restricting the development of maritime communication. However, the QoS such as latency and energy efficiency of data transmission can be optimized by RL [22, 23]. The anti-jamming scheme as proposed in [23] uses deep RL to optimize the transmit power based on the state that consists of the current locations of the vessels and UAVs without relying on a jamming model. Two dueling networks are used in this scheme to reduce the overestimation. Table 1.1 provides the summary of related works in maritime communications.

Table 1.1 Related works in maritime communications

Project/system	Frequency	Maximum coverage	Maximum rate	Application/feature
Navigational automated identification system [1–3]	HF, VHF, UHF band	740 km	130 Kbps	Voice, fax, and text
Long-term evolution (LTE) [4]	2.6 GHz	30 km	10 Mbps	Multimedia services
5G [5]	28 GHz	1 km	1 Gbps	Multimedia services
Mobile ad hoc network (MANET) [6]	27 MHz/40 MHz	70 km	1.2 Kbps	Multi-hop services
TRITON [7]	5.8 GHz	15 km	5 Mbps	Broadband services
GEO-based Inmarsat-6 [9]	2 GHz/28 GHz	100% of the Earth	60 Mbps	GEO
MEO-based O3B [10]	28 GHz	Within 45 degrees of the Equator	500 Mbps	MEO
LEO-based iridium [11]	L band	100% of the Earth	1.4 Mbps	LEO
BLUECOM+ [13]	500 MHz/800 MHz	100 km	3 Mbps	Broadband services

1.1.2 Challenges

Compared with terrestrial wireless communication systems, maritime communication has to address more severe propagation degradation, communication interference, and jamming. In addition, the RL-based communication schemes are challenged by the unreliable feedback of the communication policy from the environment due to dynamic maritime environments with narrowband connectivity [24]. With the booming of smart maritime applications such as ship detection, monitoring, and border patrol, maritime communications are expected to meet various QoS requirements in terms of the communication latency, packet loss rate, energy consumption, and throughput. In addition, security and privacy are also critical to support smart maritime applications, e.g., the authentication framework has to address spoofing attacks and man-in-the-middle attacks [25].

With the dynamic random 3D movements of the sea surface, maritime communication is challenged by the antenna misalignment and the time-variant antenna orientations and heights, which in turn results in highly dynamic channel states [7]. The wave occlusions further degenerate the channel gains and often break communication link for seconds [26]. The rough sea surface under bad weather such as storms usually results in rich scattering, large path loss, and severe shadow fading [27]. The resulting maritime channel model has to be analyzed in depth via large-scale field measurements and data mining [28].

The construction of a maritime communication system is challenging, which needs to meet ubiquitous connectivity, traffic nonuniformity, device heterogeneity, simplicity, and interoperability. For example, the maritime communication system needs to provide ubiquitous connectivity between vessels and shore, especially over open oceans including the polar regions, to ensure the unbroken and consistent existence of maritime services. Heavy traffic concentration is typical in ports, nearshore, and waterways, while the maritime traffic is highly unevenly distributed. For device heterogeneity, the maritime communication system needs to adapt to the high degree of heterogeneity of the devices for communication capabilities, including hardware and power supplies. Furthermore, maritime communication devices should be reliable and robust in the complex marine environment including harsh weather. At the same time, the system should have simplicity and low cost. At last, the maritime MTC system will provide different maritime IoT applications and services with the ability to seamlessly access the network within and across network boundaries, and to provide efficient and reliable service across the entire spectrum of maritime IoT services with no or minimal effort from end users or hosts [29, 30]. Maritime communication needs to address severe propagation degradation, communication interference and jamming, dynamic maritime environments, random sea surface, and the challenges from the booming of smart maritime applications, as illustrated in Fig. 1.2. Table 1.2 lists the challenges in maritime communications.

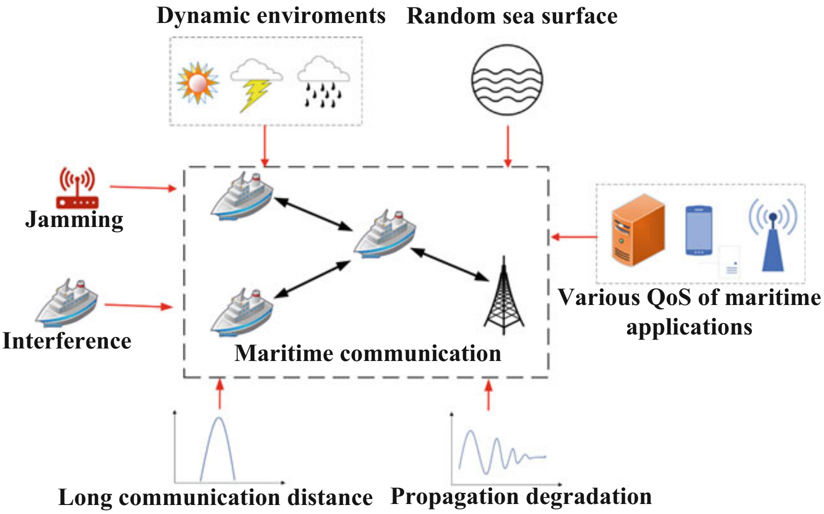


Fig. 1.2 Maritime wireless communication channels

Table 1.2 Related work on secure maritime communications

Communication environment	Attacks	Application/feature
Sea surface [27]	Interference and jamming	Various maritime applications
Dynamic maritime environments [24]	Eavesdropping	Heavy traffic concentration
Severe propagation degradation [26]	Man-in-the-middle attacks	Device heterogeneity
Antenna misalignment [7]	DoS attacks	Robust low-cost and simple system

1.1.3 Secure Maritime Communications

Maritime communications are vulnerable to attacks such as eavesdropping, spoofing, virus, and jamming [31]. For example, the Petya virus attacked the vessel tracking system of MAERSK shipping company in 2017, resulting in nearly 24-hour congestion of Port of Los Angeles and the loss of 0.3 billion dollars [32]. The secure maritime communication system as presented in [32] uses the identity-based encryption to protect the shared data among the vessels and authenticates the user request based on the unique user ID to prevent data leakage. The vessel automatic identification system as proposed in [33] that applies the time-efficient stream loss-tolerant authentication protocol to broadcast the vessel identification and position exploits compressed bloom filters to address both eavesdropping and spoofing attacks with low overhead. Maritime IoT protects data for applications such as real-time cargo status management against eavesdropping and jamming. For instance, the maritime ship data system in [34] uses the backward induction-based resource allocation and amplify-and-forward relay.

Moreover, mobile edge computing helps the maritime transportation systems handle the latency-sensitive tasks at the edge of the network with less latency and energy consumption and reduce the probability of tampering the tasks by the attackers compared with cloud computing. As the real-time tasks are offloaded to the edge for faster processing, there are also security issues such as the transmission of vessels' sensitive information in the maritime transportation systems. To secure the sensitive information transmitted and to decrease the delay and energy consumption, an IoT-based collaborative processing system can use a blockchain to protect the transmission data and a verifiable random function and reputation voting-based mechanism to reduce the communication cost in blockchain consensus communication process [35].

For the maritime industry, advanced wireless technologies such as Worldwide Interoperability for Microwave Access (WiMAX) have been applied in maritime communications to improve the data transmission performance. However, some security and privacy issues need to be addressed in these maritime communication systems. For example, WiMAX technology provides high data rate and satisfies the increasing demand of data traffic for the large capacity vessel data. To handle the authentication of a group of vessels, a WiMAX-based secure communication network can use elliptic curve Diffie-Hellman-based authentication protocol to secure the continuous data exchange between the ships and the BS or the shore with low computational complexity and overhead [25].

In addition, ML such as RL as an advanced technology can be used in maritime communication to resist attacks. In [16], a UAV-assisted maritime rescue scheme is presented, which uses Q-learning to optimize the UAV moving path against jamming attacks. To further improve the network performance, a deep RL-based maritime communication scheme is proposed in [23] to optimize the maritime transmit power and reduce the bit error rate of the message. The secure maritime transportation system in [36] applies an adaptive incremental passive-aggressive machine learning method to detect the cyberattacks and uses an approximate linear dependence and a modified hybrid forgetting mechanism to update the detection model.

In the communication between both ship-to-ship and shore-to-ship, it is important for the servers to obtain and monitor the vessels' parameters. IoT as an intelligent technique can be used in maritime transportation systems to acquire more data about the physical parameters of the vessels and share them for monitoring the vessels [37, 38]. During the transmission of IoT data between the coast-side servers and the vessels, there is a need to ensure that the data is accessed securely. In [32], an identity-based secure information sharing scheme is proposed for the maritime transport system, which uses identity-based encryption and blockchain to protect the data and improve the security of maritime transport networks. Table 1.3 provides a summary of studies on secure maritime communication systems.

1.1.4 Reliable Maritime Communications

With the rapid growth of maritime activities and the development of the maritime economy, the MF/HF/VHF maritime communication systems and long-distance and shore-ship mobile communications [24] have to meet various QoS requirements of the maritime applications against both interference and jamming in dynamic and heterogeneous networks [29]. In particular, maritime weather such as big storms severely degrade the performance of 4G or Wi-Fi systems, and maritime network has to address the interference and jamming signals sent from a larger area than terrestrial networks [39].

MEC enables data processing closer to users with less transmission latency and energy consumption and provides reliable data-driven maritime services [40, 41]. For example, the maritime MEC network as proposed in [40] integrates software-defined networks to achieve ultra-reliability, scalability, and low latency. The two-stage maritime offloading in [41] optimizes the computation and communication resource allocation for massive maritime data in terms of the low latency and energy consumption.

Table 1.3 Summary of works for secure maritime communication systems

Methods	Performance	Attacks	Systems
Q-learning [16]	High throughput	Jamming	UAV-assisted maritime rescue
Deep reinforcement learning [23]	Low bit error rate	Jamming	UAV-assisted maritime communication
Elliptic curve Diffie-Hellman-based authentication [25]	Low computational complexity and overhead	Man-in-the-middle attacks	WiMAX-based maritime communications
Identity-based encryption and blockchain [32]	High quality of service	Data leakage	IoT-enabled maritime transportation systems
Time-efficient stream loss-tolerant authentication [33]	Low communication overhead	Eavesdropping and spoofing	Maritime communication systems
Backward induction and amplify-and-forward relay [34]	High throughput	Eavesdropping and jamming	Maritime ship data systems
Blockchain [35]	Low delay and energy consumption	Replay attack and camouflage attack	Maritime IoT systems
Adaptive incremental passive-aggressive [36]	High attack detection accuracy and low latency	DDOS attacks	Maritime IoT systems

Relay enhances the communication coverage and reliability due to the spatial diversity gain [14, 18, 42–46]. For example, the relay-based maritime multicast system improves both throughput and energy efficiency for maritime equipment such as radar, sonar, and ocean sensors in [42]. The offshore mesh network in [43] chooses buoys to relay the BS messages and provide the energy source for maritime applications such as monitoring stream sensing data in marine surveillance. The high-altitude balloon-enabled maritime relay system in [44] integrates dynamic balloon networks to guarantee reliable maritime communication in a large area.

UAV-assisted relay communication systems have been widely used in maritime communications due to their deployment flexibility [14, 18]. As an example of the UAV-aided maritime communication, the caching UAV-assisted decode-and-forward relay communication system in [14] applies the one-dimensional linear search method to obtain the optimal UAV placement and thus achieves higher communication performance in the downlink maritime communication. The space-air-surface three-tier heterogeneous network constructed in [18] deploys drones as on-demand aerial access points to improve the connectivity, capacity, and flexibility for maritime wireless communications. However, the vast and complex ocean operating environment seriously degrades the communication performance of the UAV-assisted relay networks in practice, especially for meeting the all-weather and long-endurance requirements.

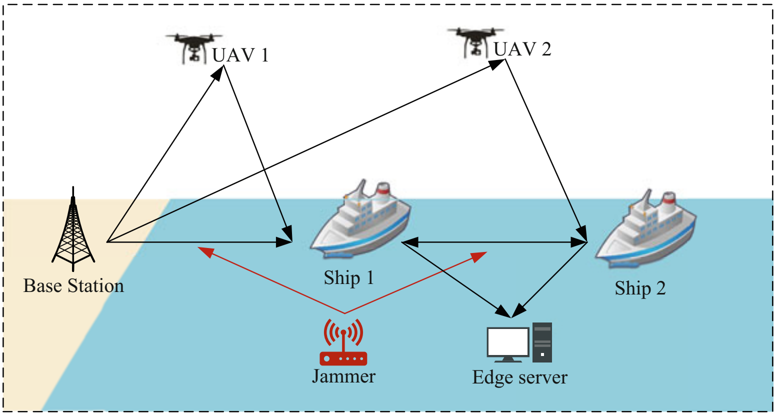


Fig. 1.3 Reliable maritime communication systems against jamming attacks and interference

Unmanned surface vehicles (USVs) with sufficient endurance and payload help improve the coverage and the transmission performance in maritime wireless communications [45, 46]. The USV-enabled maritime wireless network as proposed in [45] employed USVs to assist the communication between the terrestrial BS and ships and thus significantly improve transmission performance. In [46], a UAV-assisted cooperative transmission scheme is proposed to achieve the high-reliable and low-latency transmission in the maritime environment. Figure 1.3 shows the reliable maritime communication system model, where the system needs to reduce attacks from the jammer.

ML techniques including supervised learning, unsupervised learning, and RL have been widely applied to improve network reliability in maritime communications [47–52]. In [47], a K-means algorithm is combined with a genetic algorithm to efficiently avoid collision in dynamic vessel networks. The proactive link adaptation scheme proposed in [48] uses a nonlinear autoregressive neural network to predict the near-sea-surface channel link status and thus improve the channel utility and the link capacity for maritime communication networks. Table 1.4 provides a summary of works for reliable maritime communications.

1.2 Motivation and Objective

In this book, we investigate reliable, secure, and efficient maritime communications based on RL, which enables the maritime devices to optimize their communication and security performance in the dynamic games. For example, DQN enables maritime mobile devices to optimize the AP and edge selection based on the channel quality and the computation capacity to reduce the computation and communication costs and increase the service incomes. The corresponding deep neural network

Table 1.4 Related works on reliable maritime communications

Applications	Techniques	Performance
Seacoast BS [40]	Software-defined networking	Low latency
	Edge computing	
Coastal BS [41]	Two-stage joint optimal offloading algorithm	Low latency
		Low-energy consumption
Ship [42]	Joint beamforming optimization	High throughput
	Relay selection	Low energy consumption
Buoys [43]	Energy harvesting	Low energy consumption
	Relay selection	
Balloon [44]	Integer programming-based relay selection	Low reconfiguration frequency
		High accepted traffic
UAV [14]	One-dimensional linear search-based relay selection	High achievable rate
USV [45]	Successive convex approximation and interior-point methods	High throughput
Drone [18]	Multicast beamforming	High spectral efficiency
	Relay selection	Low required capacity
USV [46]	Many-to-one matching theory	Low latency
	Mother ship relay selection	
Vessel [47]	K-means	Low latency
Maritime IoT terminals [48]	Nonlinear autoregressive neural network	High channel utility
		High link capacity

structure, as shown in Fig. 1.4, consists of the four CNNs, each having two convolutional (Conv.) layers and four fully connected (FC) layers.

1.2.1 Learning-Based Secure Maritime Communications

RL-based secure maritime communications enable ships, maritime sensors, and satellite phones to enhance the performance to support the maritime tasks such as fish tracking and monitoring [53] against jamming, spoofing, and eavesdropping. For example, the power allocation and relay selection can apply RL algorithms such as Q-learning, Dyna-Q, post-decision state (PDS), and deep Q-network (DQN) to improve the secrecy rate and the BER.

For example, the deep RL-based UAV-assisted maritime communication scheme in [23] designs two dueling neural networks to optimize the ship transmit power and mobility policy to reduce the BER and the ship energy consumption against reactive jamming attacks. In [54], deep Q-learning-based transmission scheduling scheme designs a dual network including the main network and the target network

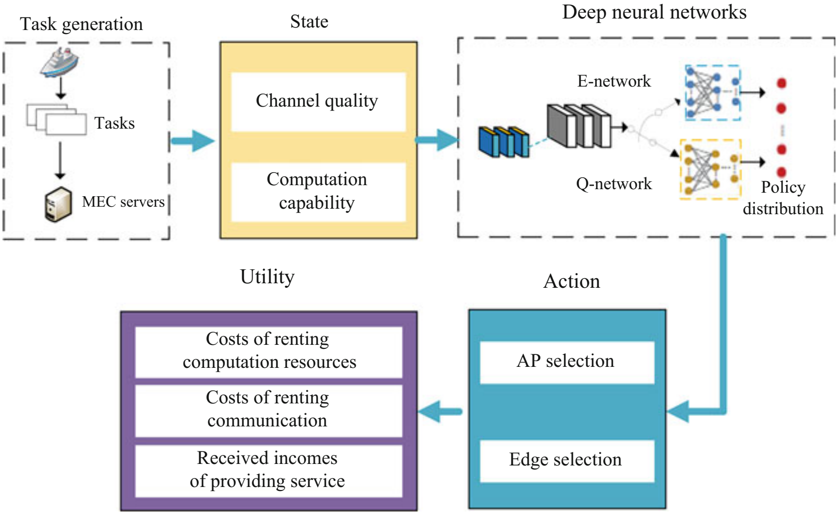


Fig. 1.4 RL-based maritime communications in [51]

to optimize data packet selection policy to reduce energy consumption and increase throughput. In [16], Q-learning-based trajectory plan scheme for cooperative search and rescue is designed to improve communication throughput.

1.2.2 Learning-Based Reliable Maritime Communications

RL-based reliable maritime communications address the dependence of the convex optimization- and Lagrangian optimization-based scheme on the accurate network and channel model. Instead, the learning-based resource allocation depends on the trial and error with the observation of the state consisting of the channel states, computation capability, and transmission quality, e.g., signal-to-interference-plus-noise ratio (SINR) obtained from the feedback control channel. The reward or utility depends on the weighted sum of the BER of the received messages, transmission latency, throughput, and energy consumption.

In the RL-based secure maritime communication as shown in Fig. 1.5, the maritime mobile device as the learning agent chooses the maritime communication policy such as transmit power, relay policy, and frequency channel. The state consists of the channel states, RSSI, BER, and jamming power, and the utility is the weighted sum of the throughput, latency, energy consumption, and computing efficiency.

For example, the formation network of unmanned ship in [49] integrates the RL strategy into the basic whale optimization algorithm to effectively improve data transfer rate, transmission latency, and network throughput in maritime broadband

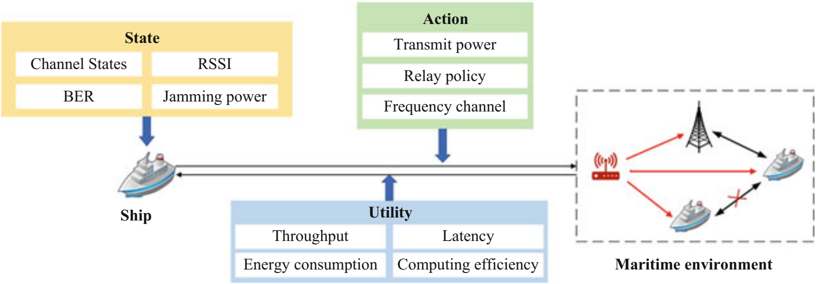


Fig. 1.5 An illustration of RL-based reliable maritime communications

Table 1.5 Learning-based maritime secure and reliable maritime communication

Agent	RL techniques	Policy	Performances
Ship [54]	DQN	Data packet selection	Low energy consumption
			High throughput
UAV [23]	DQN	Transmit power	Low BER
			Low energy consumption
Ship [49]	Q-learning	Node access	Low transmission latency
Ship [50]	Q-learning	Router selection	High packet delivery ratio
User [51]	DQN	AP and edge selection	High computing efficiency
Vessels [52]	Multi-agent DRL	Offloading rate	Low latency
UAV [16]	Q-learning	UAV trajectory	High throughput

communications. Q-learning as a model-free RL technique has been used to optimize the end-to-end delay and packet delivery ratio and thus improve the reliability of the maritime emergency video transmission in maritime search and rescue networks [50]. The DQN is applied in [51] to improve the communication and computing efficiency of maritime networks.

In [52], the vessels apply multi-agent deep RL to achieve the ultra-reliable and low-latency communications based on the channel states between the terrestrial BS and the vessels in the maritime communication networks. Table 1.5 provides the summary of learning-based maritime secure and reliable maritime communications.

1.3 Integrated Space-Air-Ground-Ocean Communication Networks

Wireless networks have evolved from the first-generation (1G) networks to the upcoming/recent 5G networks with focuses on data rate, end-to-end latency, reliability, energy efficiency, coverage, and spectrum utilization. According to the International Telecommunication Union (ITU), 5G networks have three main types of usage scenarios: enhanced mobile broadband (eMBB), ultra-reliable and low-latency communication (URLLC), and massive machine-type communications (mMTC) to account for supporting diverse services [17, 55]. In that regard, technologies including millimeter-wave (mmWave), massive multiple-input multiple-output (MIMO), and device-to-device (D2D) transmissions and so on are employed to provide users with better QoS and quality of experience (QoE), as well as to improve the network performance [17, 55], where QoS is a description of the overall performance of a service, while QoE is a measurement of the delight or annoyance of a device's experiences with a service.

While 5G networks are being deployed, people from both the industry and academia have already paid attention to the research on 6G networks [56–60], where 6G networks are expected to effectively support high-quality services, new emerging applications (e.g., virtual and augmented reality, remote surgery, and holographic projection), and unlimited connectivity for the massive number of smart terminals. For instance, the roadmap toward 6G networks is discussed [56] along with requirements, enabling techniques and architectures.

Different from previous generation networks, 6G networks will be required to revolutionize themselves by realizing intelligence to meet more stringent requirements and demands for the intelligent information society of 2030, which include [56–60] ultrahigh data rates, a peak data rate of at least 1 Tb/s, and a user-experienced data rate of 1 Gb/s; ultralow latency, less than 1 ms end-to-end delay, even 10–100 us; ultrahigh reliability, about $1–10^{-9}$; high energy efficiency (EE), on the order of 1 pJ/b; very high mobility, up to 1000 km/h; massive connection, up to 10^7 devices/km^2 and traffic capacity of up to 1 Gbs/m^2; large frequency bands (e.g., 1 THz–3 THz); and connected intelligence with AI capability.

Furthermore, in order to support near-instant and seamless super-connectivity, an integrated space-air-ground-ocean network (ISAGON) will be the core potential architecture of 6G networks [56, 57], as shown in Fig. 1.6. Note: V2X for vehicle to everything, VLC for visible light communication, RAN for radio access networks, SDN for software-defined networking, NFV for network function virtualization, PHY for physical layer, and MAC for medium access control. The objective of ISAGON is to extremely provide broad coverage and seamless connectivity for space, airborne, ground, and underwater areas, such as airplanes in the sky, ships at sea, monitors at remote areas, or vehicles on land. As a result, human activities will dramatically expand from the ground to air, space, and deep sea. At the same time, centralized and edge computing clouds are deployed at RAN with SND and NFV to provide powerful computational processing and massive data acquisition for ISAGON. The ISAGON mainly consists of the following four tiers.

The pace-network tier deploys low Earth orbit, medium Earth orbit, and geostationary Earth orbit satellites [56, 57] to provide orbit or space services for areas not covered by ground networks. Air-network tier employs various aerial platforms including UAVs, airships, and balloons associated with flying BSs to support flexible and reliable wireless connectivity for remote areas or urgent events, and the ground-network tier is the main solution for supporting diverse services for a massive number of devices. In order to satisfy various services, this layer mainly exploits low-frequency, microwave, mmWave, visible light, and terahertz (THz)

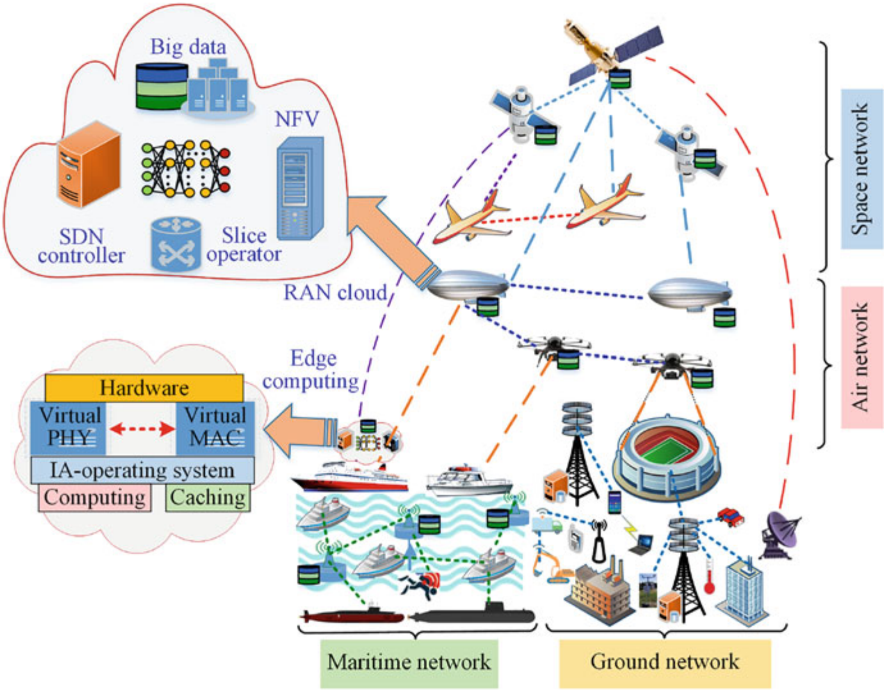


Fig. 1.6 The typical architecture of 6G network (ISAGUN)

bands for 6G networks, and the ocean-network tier aims to provide underwater communication connectivity, observation, and monitoring services for broad-sea and deep-sea activities.

According to the evolution rules of networks, initial 6G networks will be mainly supported by the existing 5G infrastructures, such as the architectures of SDN, NFV, and network slicing (NS). However, compared with 5G networks, 6G networks are required to support the abovementioned stringent requirements (e.g., ultrahigh data rates, ultralow latency, ultrahigh reliability, and seamless connectivity). At the same time, the development of 6G networks (ISAGON) has large dimension and high complexity, dynamicity, and heterogeneity characteristics. All the abovementioned issues call for a new architecture that is flexible, adaptive, agile, and intelligent. Artificial intelligence (AI) [56, 58], with strong learning ability, powerful reasoning ability, and intelligent recognition ability, allows the architecture of 6G networks to learn and adapt itself to support diverse services accordingly without human intervention. In [56, 57], AI-enabled techniques are applied to achieve network intelligentization, closed-loop optimization, and intelligent wireless communication for 6G networks. Kato et al. [56, 58] use deep learning to optimize the performance of space-air-ground-integrated networks and show how to employ deep learning to select the most suitable paths for satellite networks. Furthermore, DRL is adopted to preserve reliable wireless connectivity for UAV-enabled networks by learning the environment dynamics [10]. Simulation results demonstrated that DRL significantly outperforms conventional methods. Hence, it is promising to adopt AI to 6G networks to optimize the network architecture and improve the network performance.

Although in the studies [56, 58] AI is applied to enable intelligent wireless networks, it is yet to investigate how to systematically sense data from environments, analyze collected data, and then apply the discovered knowledge to optimize network performance for 6G. Hence, in the following, an AI-enabled intelligent architecture for 6G networks is presented to realize smart resource management, automatic network adjustment, and intelligent service provisioning with a high level of intelligence, where the architecture consists of four layers: sensing layer, data mining and analytics layer, control layer, and application layer. The proposed architecture aims at intelligently extracting valuable information from massive data, learning and supporting different functions for self-configuration, self-optimization, and self-healing in 6G networks, in order to tackle optimized physical layer design, complicated decision-making, network management, and resource optimization tasks. Based on AI-enabled intelligent 6G networks, we introduce the applications of AI techniques in the context of AI-empowered mobile edge computing, intelligent mobility and handover management, and smart spectrum management. After that, we discuss important future research directions for AI-enabled 6G intelligent networks. Finally, some conclusions are drawn.

1.3.1 AI-Enabled Intelligent Space-Air-Ground-Ocean Communication Networks

The development of ISAGON networks will be large-scale, multilayered, highly complex, dynamic, and heterogeneous. In addition, ISAGON networks need to support seamless connectivity and guarantee diverse QoS requirements of the huge number of devices, as well as process large amount of data generated from physical environments. AI techniques with powerful analysis ability, learning ability, optimizing ability, and intelligent recognition ability can be employed into ISAGON networks to intelligently carry out performance optimization, knowledge discovery, sophisticated learning, structure organization, and complicated decision-making. With the help of AI, we present an AI-enabled intelligent architecture for ISAGON networks which is mainly divided into four layers: intelligent sensing layer, data mining and analytics layer, intelligent control layer, and smart application layer, as shown in Fig. 1.7.

Here, we introduce some common AI techniques as follows. AI techniques subsume multidisciplinary techniques including machine learning (supervised learning, unsupervised learning, and reinforcement learning), deep learning, optimization theory, game theory, and meta-heuristics [58]. Among them, machine learning, deep

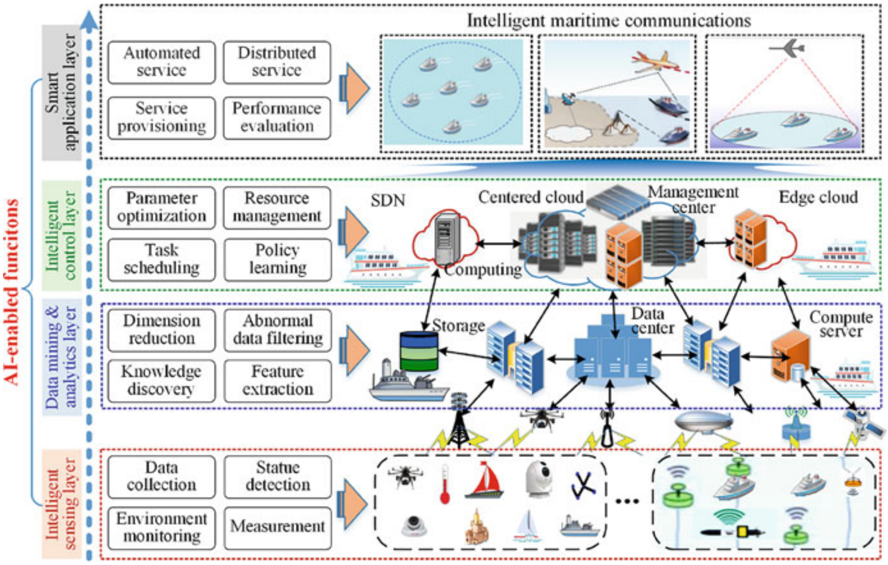


Fig. 1.7 The architecture of an AI-enabled intelligent ISAGON network

learning in particular, is the most popular AI subfield which is widely adopted in wireless networks.

Supervised Learning: Supervised learning uses a set of exclusive labeled data to build the learning model (also called training), which is broadly divided into classification and regression subfields. Classification analysis aims to assign a categorical label to every input sample, which mainly includes decision trees (DT), support vector machine (SVM), and K-nearest neighbors (KNN). Regression analysis contains support vector regression (SVR) and Gaussian process regression (GPR) algorithms, and it estimates or predicts continuous values based on the input statistical features.

Unsupervised Learning: The task of unsupervised learning is to discover hidden patterns as well as to extract the useful features from unlabeled data. It is generally divided into clustering and dimension reduction. Clustering seeks to group a set of samples into different clusters according to their similarities, and it mainly includes K-means clustering and hierarchical clustering algorithms. Dimension reduction transforms a high-dimensional data space into a low-dimensional space without losing much useful information. Principal component analysis (PCA) and isometric mapping (ISOMAP) are two classic dimension reduction algorithms.

Reinforcement Learning: In RL, each agent learns to map situations to actions and makes suitable decisions on what actions to take through interacting with the environment, so as to maximize a long-term reward. Classic RL algorithms include Markov decision process (MDP), Q-learning, policy learning, actor critic (AC), DRL, and multi-armed bandit (MRB).

Deep Learning: Deep learning is an AI function that realizes the working of the human brain in understanding the data representations and creating patterns based on artificial neural networks. It consists of multiple layers of neurons, and the learning model can be supervised, semi-supervised, and unsupervised. Classic deep learning algorithms include deep neural network (DNN), convolutional neural network (CNN), recurrent neural network (RNN), and long short-term memory (LSTM).

Intelligent Sensing Layer Generally, sensing and detection are the most primitive tasks in ISAGON networks, where ISAGON networks tend to intelligently sense and detect the data from a physical environment via enormous devices (e.g., cameras, sensors, vehicles, drones, and smartphones) or crowds of human beings. AI-enabled sensing and detecting are capable of intelligently collecting a large amount of dynamic, diverse, and scalable data by directly interfacing the physical environment, mainly including radio-frequency utilization identification, environment monitoring, spectrum sensing, intrusion detection, interference detection, and so on.

It is worth noticing that high accurate sensing, real-time sensing, and robust sensing are of great interest, since ISAGON networks need to support ultrahigh reliability and ultralow-latency communication services. In addition, dynamic ISAGON networks lead to spectrum characteristic uncertainty, which entails great difficulty in robust and accurate sensing. AI techniques can realize accurate, real-time, and robust spectrum sensing, where fuzzy SVM and nonparallel hyperplane SVM are robust to environment uncertainties, CNN-based cooperative sensing can improve sensing accuracy with low complexity, the combination of K-means clustering and SVM is capable of achieving real-time sensing by training low-dimensional input samples, and Bayesian learning can address large-scale heterogeneous sensing problems by tackling heterogeneous data fusion.

For example, in ISAGON networks, spectrum sensing is an important technique to improve the spectrum usage efficiency and address spectrum scarcity problems. However, spectrum sensing in large-scale ISAGON networks is very challenging since a massive number of devices aim to sense spectrum simultaneously, and the massive spectrum usage detections lead to high-dimensional search problems. In this case, AI technologies can be applied to identify the spectrum characteristics and intelligently establish suitable training models to sense spectrum working status. In detail, AI-enabled learning models (e.g., SVM and DNN) detect the spectrum working status by categorizing each feature vector (spectrum) into either of the two classes, namely, the "spectrum idle class" and "spectrum buy class," and adaptively update the learning models based on dynamic environments.

Data Mining and Analytics Layer This layer has a core task that aims to process and analyze a massive amount of raw data generated from the huge number of devices in ISAGON networks and achieve semantic derivation and knowledge discovery. The massive collected data from physical environments may be heterogeneous, nonlinear, and high dimensional, so data mining and analytics can be applied in ISAGON networks to address the challenges of processing the massive amount of data, as well as to analyze the collected data toward knowledge discovery.

On the one hand, it is costly to transmit or store the massive raw data in dense networks. Hence, it is necessary to reduce data dimension of the raw data, filter abnormal data, and finally achieve a more reasonable dataset. In AI-based data mining, PCA and ISOMAP are the two common AI algorithms that can help ISAGON networks to transform higher-dimensional data into a lower-dimensional subspace [56, 58], to dramatically decrease the computing time, storage space, and model complexity. For example, the massive collected channel information, traffic flows, images, and videos from sensors, devices, or social media are high-dimensional data, which can be compressed into a small set of useful variables of the raw data by using PCA or ISOMAP. In addition, the collected data from physical environments also have abnormal data (e.g., interference, incomplete, and useless data), which can be filtered by utilizing PCA or ISOMAP.

On the other hand, data analytics is responsible for intelligently analyzing the collected data to discover useful information and form valuable knowledge. In ISAGON networks, massive data are collected from physical environment, cyber world, and social network which contain valuable information and meaningful features. Data analytics have brought us an excellent opportunity to understand the essential characteristics of wireless networks and achieve more clear and in-depth knowledge of the behavior of ISAGON networks; finally valuable patterns or rules can be discovered as knowledge to provide suitable solutions for resource management, protocol adaptation, architecture slicing, cloud computing, signal processing, and so on. For instance, based on the discovered knowledge, ISAGON is able to efficiently understand the mobility patterns of UAVs in the sky, establish the channel path loss model of a satellite-ground link, and predict the device behavior in ground networks.

Intelligent Control Layer Briefly, intelligent control layer mainly consists of learning, optimization, and decision-making. It utilizes the appropriate knowledge from lower layers to enable massive agents (e.g., devices and BSs) to smartly learn, optimize, and choose the most suitable actions (e.g., power control, spectrum access, routing management, and network association), with dual functions to support diverse services for social networks. Such function is realized by applying AI techniques in ISAGON networks, where each agent is equipped with an intelligent brain (learning model) to automatically learn to make decisions by itself.

Learning is a process of utilizing or modifying existing knowledge and experience to improve the behavior of each device or service center, so that ISAGON networks can intelligently realize optimal network slicing, end-to-end PHY design, edge computing, resource management, heterogeneous network design, and so on, according to different requirements of applications. Intelligence is an important characteristic of ISAGON networks, where the combination of AI and ISAGON networks can learn to achieve self-configuration, self-optimization, self-organization, and self-healing, finally increasing the feasibility level. For instance, post-massive multiple-input multiple-output (PM-MIMO) will be employed in ISAGON networks to support hundreds or thousands of transmit/receive antennas with mmWave or THz transmissions [56]. How to achieve optimal energy-efficient beamforming variables and mitigate RF nonlinearities is challenging. An RNN-based solution has the ability to capture the nonlinearities of RF components, where RNN learns the nonlinearities of power amplifier arrays and optimizes minimal transmitted power levels at transmitters [11].

The task of optimization is to run a deterministic and rule-based algorithm with parameter optimization based on global objectives (e.g., QoS, QoE, connectivity, and coverage). Traditional optimization algorithms (e.g., Lagrangian duality and gradient methods) have heavy mathematical models which may not be suitable for ISAGON networks, since ISAGON networks will be significantly dynamic and complex. In AI-enabled intelligent ISAGON networks, network parameters and architectures can be optimized through AI techniques, instead of traditional tedious computation. AI techniques provide the best opportunity to train auto-learning models to realize network optimization for ISAGON wireless networks, allowing providers or operators to optimize the network parameters, resources, or architectures to better adapt services for devices, making ISAGON networks intelligent, agile, and adapting. For instance, deep learning can enable SDN/NFV into an intelligent network architecture with fast learning, quick adaptation, and self-healing, which is capable of quickly optimizing network parameters and architectures to achieve intelligent softwarization, cloudization, virtualization, and slicing.

Decision-making is an important cognitive task that enables massive agents to intelligently reason, plan, and choose the most suitable decisions to meet the high-quality service requirements. In decision-making, each agent simultaneously attempts to obtain the new knowledge of other machines (called "exploration") and selects the global actions with the highest benefit based on existing knowledge (called "exploitation"). The objective of decision-making is common in ISAGON networks, e.g., selecting the optimal precoding variables in mmWave or THz transceiver systems, choosing the suitable routing strategy for dynamic ISAGON, and selecting the flexible spectrum management framework for massive multi-access scenario; all these decision-making issues can be effectively achieved by using AI techniques (e.g., MDP, game theory, RL, and optimization theory).

Smart Application Layer The main responsibilities of this layer are to deliver application-specific services to the human beings according to their colorful requirements and to evaluate the provisioned services before feedbacking the evaluation results to the intelligence process. Intelligent programming and management can be achieved by the impetus of AI to support various high-level smart applications, such as automated services, smart city, smart industry, smart transportation, smart grid, and smart health and to handle global management relevant to all smart-type applications. All activities of smart devices, terminals, and infrastructures in ISAGON networks are managed by the smart application layer through the AI techniques to realize network self-organization ability.

Another objective of this layer is to evaluate the service performance, where lots of considerations and factors can be involved, such as QoS, QoE, quality of collected data, and quality of learned knowledge. At the same time, the cost dimension metric in terms of resource efficiency is required to be taken into account, such as spectrum utilization efficiency, computational efficiency, energy efficiency, and storage efficiency. All the abovementioned evaluation metrics can be utilized to improve intelligent resource management, automatic network slicing, and smart service provisioning.

1.3.2 AI Techniques for Maritime Communication Networks

In this section, we elaborate how the AI techniques grant preliminary intelligence for maritime communication networks in terms of AI-empowered mobile edge computing, intelligent mobility and handover management, and smart spectrum management.

AI-Empowered Mobile Edge Computing MEC will be an important enabling technology for the emerging ISAGON networks, where it provides computing, management, and analytics facilities inside RAN or SDN in close proximity to various devices. In MEC, the decision-making optimization, knowledge discovery, and pattern learning are sophisticated due to the multidimensional, randomly uncertain, and dynamic characteristics. Hence, traditional algorithms (e.g., Lagrangian duality) may face a limitation in such complex networks. AI techniques can extract valuable information from collected data, learning and supporting different functions for optimization, prediction, and decision in MEC. Figure 1.8 shows the framework of AI-empowered mobile edge computing, which consists of central cloud computing and edge computing.

In edge computing servers, due to the limited capability, lightweight AI algorithms can be utilized to provide smart applications for edge scenarios (e.g., transportation and agriculture), as shown in Fig. 1.8b. For example, RL-based edge computing resource management is a model-free scheme which does not need historical knowledge and can be used to learn the environment dynamics and make suitable control decisions in real time. In the RL framework, at each step, after obtaining the state (e.g., device mobility, requirement dynamics, and resource condition) by interacting with environments, the possible resource management solutions (e.g., energy management, resource allocation, and task scheduling) are contained into the set of possible actions. Each RL agent (e.g., device or service center) selects the best action from a set of possible actions or chooses one action randomly to maximize its reward, where the reward can be determined by data rate, latency, reliability, and so on.

In the central cloud with powerful computation capability, complex centralized large-scale AI algorithms can be employed to provide various learning functions, as shown in Fig. 1.8a. For instance, as service applications in MEC networks are

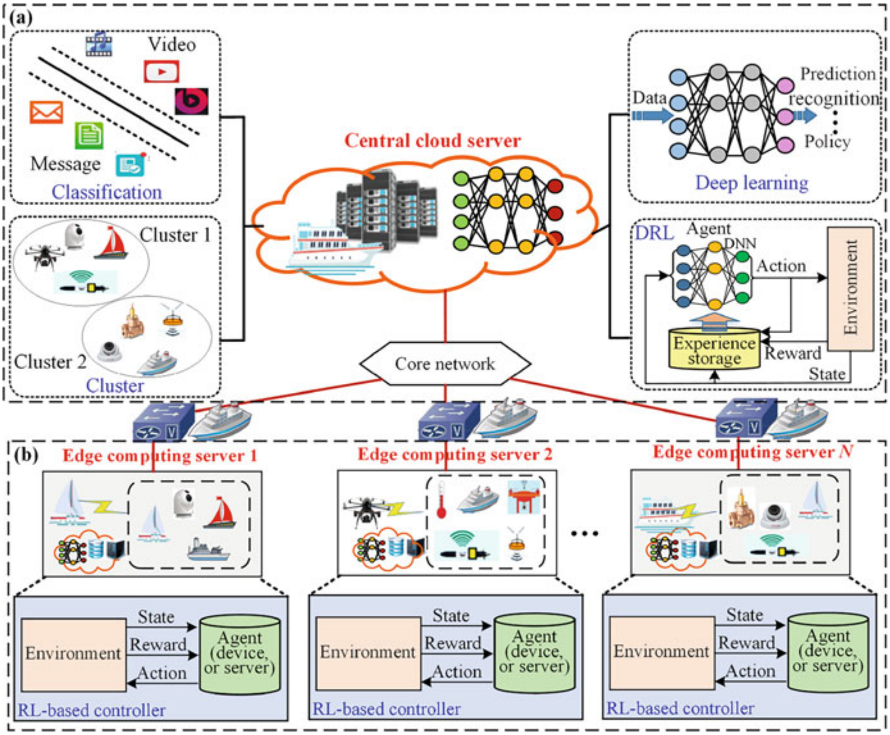


Fig. 1.8 The framework of AI-empowered MEC

diverse and dynamic, AI-based classification can be used to efficiently customize traffic flow decision for various service features. In addition, MEC server association can be obtained by AI-based cluster instead of individual decision, which will be more effective in reducing the number of participants. A central cloud server may receive massive data from edge computing servers and the data need to be trained to automatically extract features and discover knowledge. In this case, deep learning can be adopted to train computational models to achieve service recognition, traffic and behavior prediction, security detection, and so on. Moreover, in complex and dynamic MEC, the mapping between resource management decisions and the effect on the physical environments is not easy to be analytically defined. DRL can be adopted to search the optimal resource management policy under high-dimensional observation spaces. Experience replay is also adopted in DRL to utilize the historical knowledge to improve learning efficiency and accuracy, allowing the MEC to support high-quality services for edge devices.

Intelligent Mobility and Handover Management Mobility and handover management are probably the two most challenging issues of ISAGON networks, since ISAGON networks are highly dynamic, multilayer, and high dimensional, leading to frequent handovers. AI techniques can be adopted to intelligently achieve mobility

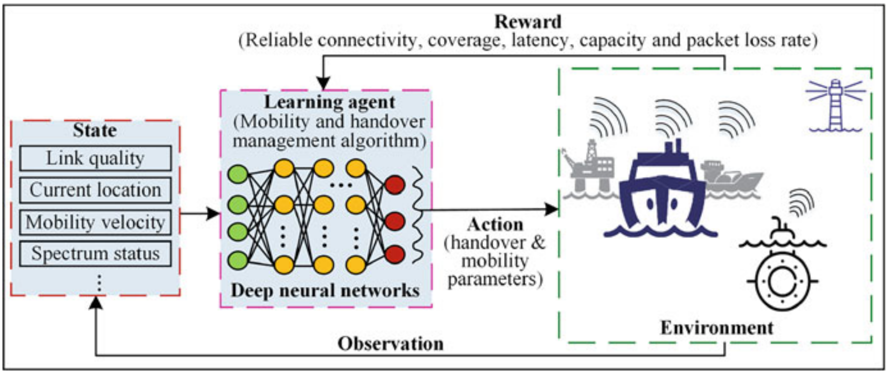

Reward
(Reliable connectivity, coverage, latency, capacity and packet loss rate)
Learning agent
(Mobility and handover management algorithm)
State
Link quality
Current location
Mobility velocity
Spectrum status
⋮
Deep neural networks
Action
(handover & mobility parameters)
Environment
Observation

Fig. 1.9 DRL in the context of mobility and handover management

prediction and optimal handover solutions to guarantee communication connectivity [61].

For example, UAV communications will be integrated into ISAGON networks, and the high-speed mobility of UAVs may lead to frequent handovers. In addition, the diverse service requirements in terms of high data rate, high reliability, and low latency increase the difficulty in processing efficient handover. At the same time, the high mobility of devices and UAVs results in uncertainties of their locations. One of the AI techniques, namely, DRL (DRL combines DL with RL to learn itself from experience), is capable of solving complex decision-making tasks, which learns to optimize the handover strategies in real time by exhibiting dynamic temporal mobility behaviors of devices or UAVs in an online manner while minimizing the transmission latency and guaranteeing reliable wireless connectivity [61]. Figure 1.9 shows the context of intelligent mobility and handover management based on DRL for UAV-enabled networks, where each UAV can be regarded as a learning agent to learn management policy by interacting with its environments. Each agent senses the environment states (e.g., link quality, current location, and velocity) and discovers the most suitable actions (e.g., mobility and handover parameters) to obtain the maximal reward, where the reward can be determined by communication connectivity, latency, capacity, and so on. In the DRL framework, UAVs can learn how to move and hand over automatically and robustly, how to reduce the latency and the handover failure probability, and finally, how to provide better services for ground devices.

ISAGON networks need to meet high-speed mobility and delay-sensitive requirements of vehicles in large-scale vehicular networks, so efficient mobility management is a key evaluation to satisfy reliability, continuity, and low latency requirements of vehicular communications. Deep learning (such as RNN and ANN)-based predictive mobility management and fuzzy Q-learning-based handover parameter optimization can learn the mobility patterns of high-speed vehicular users, which can effectively avoid frequent handovers, handover failures, or connectivity failures [61]. In addition, LSTM is a powerful AI tool for solving handover problems, as it exploits both the previous and future mobility contexts of vehicles for learning a sequence of future time-dependent movement states and predicts vehicles' trajectories to optimize handover parameters to avoid frequent handovers.

1.4 Major Contributions and Structural Arrangements

This book investigates RL-based secure and reliable maritime communications which provide a broad coverage of the maritime wireless communication issues, such as reliability, security, resource management, and privacy protection. This book consists of four rigorously refereed research topics as follows.

Chapter 2 presents the system model and problem formulation of IRS-aided maritime wireless communications. The RL-based solution and performance evaluations are also provided.

Chapter 3 mainly introduces the related works and challenges of privacy protection during the offloading process in maritime scenarios. In addition, we propose an RL-based privacy-aware offloading scheme to enable IoT devices to protect both the user location and usage pattern privacy.

Chapter 4 proposes an RL-based resource management algorithm for ultra-reliable low-latency maritime communications. The algorithm design and simulation results are provided in this chapter.

Chapter 5 mainly introduces the related works and challenges of location privacy protection in the location-based service system. In addition, we propose an RL-based-sensitive semantic location privacy protection scheme. This scheme uses differential privacy technique to randomize the released vehicle locations and adaptively selects the perturbation policy to improve the privacy protection performance.

Finally, we draw conclusions of this book in Chap. 6 and identify the future research directions.

References

1. S.C. Systems, PACTOR-4. https://www.p4dragon.com/pactor-4.html
2. S. Bauk, Chapter 12. A review of NAVDAT and VDES as upgrades of maritime communication systems, *Advances in Marine Navigation and Safety of Sea Transportation* (CRC, Jun. 2019), pp. 81–82. ISBN:978-0-42934-193-9
3. M.A. Cervera, A. Ginesi, K. Eckstein, Satellite-based vessel automatic identification system: A feasibility and performance analysis. Int. J. Satellite Commun. Netw. **29**(2), 117–142 (2011)
4. S.-W. Jo, W.-S. Shim, LTE-maritime: High-speed maritime wireless communication based on LTE technology. IEEE Access **7**, 53172–53181 (2019)

5. J. Mashino, K. Tateishi, K. Muraoka, Maritime 5G experiment in windsurfing world cup by using 28 GHz band massive MIMO, in *Proceedings of the IEEE International Symposium on Personal, Indoor Mobile Radio Communications*, Bologna, Italy, Sep. 2018
6. T. Yoshikawa, S. Kawasaki, M.e.a. Takase, Development of 27MHz/40MHz bands maritime wireless ad-hoc networks, in *Proceedings of the International Conference on Ubiquitous Future Networks*, Jeju, South Korea, Jun. 2010
7. M.-T. Zhou, V.D. Hoang, H.e.a. Harada, TRITON: High-speed maritime wireless mesh network. IEEE Wireless Commun. **20**(5), 134–142 (2013)
8. F. Bekkadal, Emerging maritime communications technologies, in *Proceedings of the International Conference on Intelligent Transport System Telecommunications*, Lille, France, Oct. 2009
9. Imnmarsat, I-6 SATELLITES. https://www.inmarsat.com/en/about/technology/satellites.html
10. G. Giambene, S. Kota, P. Pillai, Satellite-5G integration: a network perspective. IEEE Netw. **32**(5), 25–31 (2018)
11. K. Sekiguchi, Iridium contributes to "maritime safety", in *Proceedings of the Techno-Ocean*, Kobe, Japan, Oct. 2016
12. X. Li, W. Feng, J. Wang, Y. Chen, N. Ge, C.-X. Wang, Enabling 5G on the ocean: A hybrid satellite-UAV-terrestrial network solution. IEEE Wireless Commun. **27**(6), 116–121 (2020)
13. R. Campos, T. Oliveira, N. Cruz, A. Matos, J.M. Almeida, BLUECOM+: Cost-effective broadband communications at remote ocean areas, in *Proceedings of the Oceans*, Shanghai, China, Apr. 2016
14. J. Zhang, F. Liang, B. Li, Z. Yang, Y. Wu, H. Zhu, Placement optimization of caching UAV-assisted mobile relay maritime communication. China Commun. **17**(8), 209–219 (2020)
15. A. Brown, D. Anderson, Trajectory optimization for high-altitude long-endurance uav maritime radar surveillance. IEEE Trans. Aerosp. Electron. Syst. **56**(3), 2406–2421 (2019)
16. T. Yang, Z. Jiang, R. Sun, N. Cheng, H. Feng, Maritime search and rescue based on group mobile computing for unmanned aerial vehicles and unmanned surface vehicles. IEEE Trans. Ind. Inf. **16**(12), 7700–7708 (2020)
17. T. Wei, W. Feng, Y. Chen, C.-X. Wang, N. Ge, J. Lu, Hybrid satellite-terrestrial communication networks for the maritime internet of things: Key technologies, opportunities, and challenges. IEEE Internet Things J. **8**(11), 8910–8934 (2021)
18. S. Guan, J. Wang, C. Jiang, R. Duan, Y. Ren, T.Q. Quek, MagicNet: The maritime giant cellular network. IEEE Commun. Mag. **59**(3), 117–123 (2021)
19. Y. Wang, W. Feng, J. Wang, T.Q. Quek, Hybrid satellite-UAV-terrestrial networks for 6G ubiquitous coverage: A maritime communications perspective. IEEE J. Sel. Areas Commun. **39**(11), 3475–3490 (2021)
20. T. Yang, J. Chen, N. Zhang, AI-empowered maritime internet of things: a parallel-network-driven approach. IEEE Netw. **34**(5), 54–59 (2020)
21. X. Ji, J. Wang, Y. Li, Q. Sun, C. Xu, Modulation recognition in maritime multipath channels: A blind equalization-aided deep learning approach. China Commun. **17**(3), 12–25 (2020)
22. Z. Wang, B. Lin, L. Sun, Y. Wang, Intelligent task offloading for 6G-enabled maritime IoT based on reinforcement learning, in *Proceedings of the International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, Chengdu, China, Sep. 2021
23. K. Liu, P. Li, C. Liu, L. Xiao, L. Jia, UAV-aided anti-jamming maritime communications: A deep reinforcement learning approach, in *Proceedings of the 13th International Conference on Wireless Communications and Signal Processing (WCSP)*, Changsha, China, Dec. 2021
24. D. Kidston, T. Kunz, Challenges and opportunities in managing maritime networks. IEEE Commun. Mag. **46**(10), 162–168 (2008)
25. T. Yang, C. Lai, R. Lu, R. Jiang, EAPSG: Efficient authentication protocol for secure group communications in maritime wideband communication networks. Peer Peer Netw. Appl. **8**(2), 216–228 (2015)
26. C.-W. Ang, S. Wen, Signal strength sensitivity and its effects on routing in maritime wireless networks, in *Proceedings of the Conference on Local Computer Networks (LCN)*, Montreal, Canada, Oct. 2008

27. P.-Y. Kong, H. Wang, Y. Ge, C.-W. Ang, S. Wen, J.S. Pathmasuntharam, M.-T. Zhou, H.V. Dien, A performance comparison of routing protocols for maritime wireless mesh networks, in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Las Vegas, NV, Mar. 2008
28. X. Huang, K. Wu, M. Jiang, L. Huang, J. Xu, Distributed resource allocation for general energy efficiency maximization in offshore maritime Device-to-Device communication. IEEE Wireless Commun. Lett. **10**(6), 1344–1348 (2021)
29. T. Xia, M.M. Wang, J. Zhang, L. Wang, Maritime internet of things: Challenges and solutions. IEEE Wireless Commun. **27**(2), 188–196 (2020)
30. M.M. Wang, J. Zhang, X. You, Machine-type communication for maritime internet of things: A design. IEEE Commun. Surv. Tuts. **22**(4), 2550–2585 (2020)
31. G. Kavallieratos, V. Diamantopoulou, S.K. Katsikas, Shipping 4.0: Security requirements for the cyber-enabled ship. IEEE Trans. Ind. Inf. **16**(10), 6617–6625 (2020)
32. B.B. Gupta, A. Gaurav, C.-H. Hsu, B. Jiao, Identity-based authentication mechanism for secure information sharing in the maritime transport system. IEEE Trans. Intell. Transp. Syst. **24**(2), 1–9 (2021)
33. S. Sciancalepore, P. Tedeschi, A. Aziz, R. Di Pietro, Auth-ais: secure, flexible, and backward-compatible authentication of vessels ais broadcasts. IEEE Trans. Dependable Secure Comput. **19**(4), 2709–2726 (2021)
34. T. Yang, H. Feng, C. Yang, R. Deng, G. Guo, T. Li, Resource allocation in cooperative cognitive radio networks towards secure communications for maritime big data systems. Peer Peer Netw. Appl. **11**(2), 265–276 (2016)
35. P. Zhang, Y. Wang, G.S. Aujla, A. Jindal, Y.D. Al-Otaibi, A blockchain-based authentication scheme and secure architecture for iot-enabled maritime transportation systems. IEEE Trans. Intell. Transp. Syst. **24**(2), 2322–2331 (2022)
36. E. Gyamfi, J.A. Ansere, M. Kamal, M. Tariq, A. Jurcut, An adaptive network security system for IoT-enabled maritime transportation. IEEE Trans. Intell. Transp. Syst. **24**(2), 2538–2547 (2022)
37. Y. Huo, X. Dong, S. Beatty, Cellular communications in ocean waves for maritime internet of things. IEEE Internet Things J. **7**(10), 9965–9979 (2020)
38. 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), 5374–5385 (2020)
39. J. Liu, Y. Shi, Z.M. Fadlullah, N. Kato, Space-air-ground integrated network: A survey. IEEE Commun. Surv. Tuts. **20**(4), 2714–2741 (2018)
40. X. Su, L. Meng, J. Huang, Intelligent maritime networking with edge services and computing capability. IEEE Trans. Veh. Technol. **69**(11), 13,606–13,620 (2020)
41. T. Yang, H. Feng, S. Gao, Z. Jiang, M. Qin, N. Cheng, L. Bai, Two-stage offloading optimization for energy-latency tradeoff with mobile edge computing in maritime internet of things. IEEE Internet Things J. **7**(7), 5954–5963 (2019)
42. R. Duan, J. Wang, H. Zhang, Y. Ren, L. Hanzo, Joint multicast beamforming and relay design for maritime communication systems. IEEE Trans. Green Commun. Netw. **4**(1), 139–151 (2019)
43. A. Hosseini-Fahraji, P. Loghmannia, K. Zeng, X. Li, S. Yu, S. Sun, D. Wang, Y. Yang, M. Manteghi, L. Zuo, Energy harvesting long-range marine communication, in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, Toronto, Canada, Aug. 2020
44. T. Ge, Y. Wang, C. Zhang, Y. Fang, Reconfiguration in maritime networks integrated with dynamic high altitude balloons, in *Proceedings of the IEEE International Conference on Communications (ICC)*, Montreal, Canada, Aug. 2021
45. C. Zeng, J.-B. Wang, C. Ding, H. Zhang, M. Lin, J. Cheng, Joint optimization of trajectory and communication resource allocation for unmanned surface vehicle enabled maritime wireless networks. IEEE Trans. Commun. **69**(12), 8100–8115 (2021)

46. L. Lyu, Y. Dai, N. Cheng, S. Zhu, X. Guan, B. Lin, X. Shen, AoI-aware co-design of cooperative transmission and state estimation for marine IoT systems. IEEE Internet Things J. **8**(10), 7889–7901 (2020)
47. T. Yang, C. Han, M. Qin, C. Huang, Learning-aided intelligent cooperative collision avoidance mechanism in dynamic vessel networks. IEEE Trans. Cogn. Commun. Netw. **6**(1), 74–82 (2019)
48. W. Xu, H. Zhou, T. Yang, H. Wu, S. Guo, Proactive link adaptation for marine internet of things in TV white space, in *Proceedings of the IEEE International Conference on Communications (ICC)*, Jun. 2020
49. H. Cao, T. Yang, Z. Yin, X. Sun, D. Li, Topological optimization algorithm for HAP assisted multi-unmanned ships communication, in *Proceedings of the IEEE Conference on Vehicular Technology (VTC2020-Fall)*, Victoria, Canada, Nov. 2020
50. Z. Wang, B. Lin, Q-learning based delay sensitive routing protocol for maritime search and rescue networks, in *Proceedings of the IEEE Conference on Vehicular Technology (VTC2020-Fall)*, Victoria, Canada, Nov. 2020
51. F. Xu, F. Yang, C. Zhao, S. Wu, Deep reinforcement learning based joint edge resource management in maritime network. China Commun. **17**(5), 211–222 (2020)
52. J. Zeng, J. Sun, B. Wu, X. Su, Mobile edge communications, computing, and caching (MEC3) technology in the maritime communication network. China Commun. **17**(5), 223–234 (2020)
53. H. Boche, R.F. Schaefer, H.V. Poor, Denial-of-service attacks on communication systems: Detectability and jammer knowledge. IEEE Trans. Signal Process. **68**, 3754–3768 (2020)
54. T. Yang, J. Li, H. Feng, N. Cheng, W. Guan, A novel transmission scheduling based on deep reinforcement learning in software-defined maritime communication networks. IEEE Trans. Cogn. Commun. Netw **5**(4), 1155–1166 (2019)
55. J.G. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, J.C. Zhang, What will 5G be? IEEE J. Sel. Areas Commun. **32**(6), 1065–1082 (2014)
56. K. David, H. Berndt, 6G vision and requirements: Is there any need for beyond 5G? IEEE Vehic. Teh. Mag. **13**(3), 72–80 (2018)
57. P. Yang, Y. Xiao, M. Xiao, S. Li, 6G wireless communications: Vision and potential techniques. IEEE Netw. **33**(4), 70–75 (2019)
58. K.B. Letaief, W. Chen, Y. Shi, J. Zhang, Y. Zhang, The roadmap to 6G: AI empowered wireless networks. IEEE Commun. Mag. **57**(8), 84–90 (2019)
59. Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, 6G wireless networks: Vision, requirements, architecture, and key technologies. IEEE Vehic. Teh. Mag. **14**(3), 28–41 (2019)
60. M.G. Kibria, K. Nguyen, G.P. Villardi, O. Zhao, K. Ishizu, F. Kojima, Big bata analytics, machine learning, and artificial intelligence in next-generation wireless networks. IEEE Access **6**, 32328–32338 (2018)
61. H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, K. Wu, Artificial-intelligence-enabled intelligent 6G networks. IEEE Netw. **34**(6), 272–280 (2020)

Chapter 2
Learning-Based Intelligent Reflecting Surface-Aided Secure Maritime Communications

Physical layer security (PLS) has attracted increasing attention as an alternative to cryptography-based techniques for maritime wireless communications [1]. For instance, secure communication services in [1, 2] exploit the wireless channel features to address eavesdropping without relying on shared secret keys. So far, a variety of approaches have been reported to improve security in wireless communication systems, which can be used in maritime wireless communications, e.g., cooperative relaying strategies [3, 4], artificial noise-assisted beamforming [5, 6], and cooperative jamming [7, 8]. However, employing a large number of antennas and relays in PLS systems incurs excessive hardware costs and high system complexity. Moreover, cooperative jamming and transmitting artificial noise require extra transmit power to guarantee transmission, and thus raise challenges to implement in maritime wireless communication systems.

To tackle these challenges of the existing approaches in [3–8], a new paradigm, called intelligent reflecting surface (IRS) [9–13], has been proposed as a promising technique to increase spectrum efficiency and energy efficiency and enhance secrecy rate in the 5G and beyond wireless communication systems. In particular, IRS is a uniform planar array consisting of low-cost passive reflecting elements, and each IRS element adaptively adjusts its reflection amplitude and/or phase to control the strength and direction of the electromagnetic wave. Hence, IRS is capable of enhancing and/or weakening the reflected signals at different users [3]. The reflected signal by IRS can increase the received signal at legitimate users while suppressing the signal at the eavesdropper ship [9–13]. Based on the innovative studies devoted to performance optimization for IRS-aided secure communications [14–25], we discuss the RL-based IRS-aided secure maritime wireless communications against eavesdropping in this chapter.

L. Xiao et al., *Reinforcement Learning for Maritime Communications*, Wireless Networks, https://doi.org/10.1007/978-3-031-32138-2_2

2.1 Related Work

Initial studies on IRS-aided secure communication systems have been reported in [14–17], which assume that a simple system model with only a single-antenna legitimate user and a single-antenna eavesdropper ship. In particular, in [14] and [15], alternative optimization (AO) is applied to jointly optimize the transmit beamforming vector at the BS and the phase elements at the IRS to maximize the secrecy rate for the single-user IRS-assisted wireless communication systems.

In recent years, both AO and semidefinite programming (SDP) relaxation is applied in the power allocation and the IRS reflecting beamforming in [18] to save the transmit power at the AP subject to the secrecy rate constraint. In addition, the secure IRS transmission framework is studied in [19] to reduce the transmit power for rank-one and full-rank AP-IRS links, and a closed-form expression of the beamforming matrix is derived.

Based on the secure communication against single eavesdropping in [14–19], secure communications against multiple eavesdroppers were investigated in [20–22]. For instance, a minimum-secrecy-rate maximization-based secure IRS-aided multiuser multiple-input single-output (MISO) communication system against multiple eavesdroppers is presented in [20]. However, the simplified system model in the optimization problem sometimes degrades the performance.

In addition, in [23] and [24], an IRS-aided MIMO system applies the suboptimal secrecy rate maximization to choose the beamforming policy against eavesdroppers with multiple antennas and the minorization-maximization (MM) algorithm to jointly optimize the AP beamforming and the IRS phase shift coefficients.

Moreover, the authors in [22] and [25] employed the artificial noise-aided beamforming for IRS-aided MISO communication systems to improve the secrecy rate. An AO-based solution was applied to jointly optimize the AP's beamforming, artificial noise interference vector, and IRS's reflecting beamforming to maximize the secrecy rate. Existing studies in [14–20, 22–25] assume perfect channel state information (CSI) of legitimate users or eavesdropper ship available at the AP. The assumption is not practical in most dynamic maritime communications, because perfect CSI is challenging to obtain at the AP and the estimated CSI usually has a large error in the time-varying channel due to the transmission and processing delay, as well as high mobility of users. Hence, in [21], an IRS-aided secure communication optimization problem with outdated CSI of the eavesdropping channels is investigated, and a robust solution against multiple eavesdropper ships is proposed.

The above studies presented in [14–25] mainly applied the traditional optimization techniques, e.g., AO, SDP, or MM algorithms, to optimize the AP's beamforming and the IRS's reflecting beamforming for secure communications, which suffer from severe performance degradation in large-scale systems. Inspired by the recent advances of artificial intelligence, several works attempted to utilize AI algorithms to optimize IRS's reflecting beamforming [26–29]. In particular, DL has been exploited to search the IRS reflection matrices that maximize the achievable system rate, and simulation results demonstrate the significant performance gain over conventional optimization algorithms. In [28] and [29], the DRL-based approach that addresses the non-convex optimization problem of the phase shifts at the IRS is proposed.

However, the works [26–29] aim to maximize the system achievable rate of a single ship, but the multiple-ship secure communication with imperfect CSI has to be further explored. In [30] and [31], an RL-based smart AP beamforming aided by IRS against an eavesdropper ship in complex environments is proposed, but the IRS's reflecting beamforming has to be jointly optimized with the AP's transmit beamforming.

In this chapter, we investigate an IRS-aided secure communication system to maximize the secrecy rate of multiple legitimate ships against multiple eavesdropper ships in time-varying maritime wireless channels and guarantee the QoS requirements of legitimate ships. A novel DRL-based secure beamforming approach is presented to jointly optimize the AP beamforming matrix, and the IRS reflecting beamforming matrix (reflection phases) in dynamic environments, with major contributions summarized as follows:

- The physical secure communication based on IRS with multiple eavesdropper ships is investigated under the time-varying channel states. In addition, we formulate a joint AP's transmit beamforming and IRS's reflecting beamforming optimization problem to maximize the system secrecy rate and satisfy the QoS requirements of legitimate ships.
- An RL-based intelligent beamforming framework is presented to achieve the optimal AP's beamforming and the IRS's reflecting beamforming, where the central controller optimizes the beamforming policy according to the instantaneous observations from the dynamic environment. Specifically, a QoS-aware reward function is constructed to cover both the secrecy rate and ships' QoS requirements into the learning process.
- A DRL-based secure beamforming approach is proposed to exploit the information of complex structure of the beamforming policy domain and improve the learning efficiency and secrecy performance. This approach designs a modified PDS learning structure to trace the channel dynamic against channel uncertainty and prioritizes experience replay (PER) to enhance the learning efficiency.
- Simulation results are discussed to demonstrate the effectiveness of the deep learning-based secure beamforming approach in terms of the secrecy rate and the QoS satisfaction, compared with the approach [14] in time-varying channel conditions.

The rest of this chapter is organized as follows. Section 2.2 presents the system model and problem formulation. The optimization problem is formulated as an RL problem in Sect. 2.3. Section 2.4 proposes a deep PDS-PER-based secure beamforming approach. Section 2.5 provides simulation results, and Sect. 2.6 concludes the chapter.

Notations: In this chapter, vectors and matrices are represented by boldface lowercase and uppercase letters, respectively. $\mathrm{Tr}(\cdot)$, $(\cdot)^*$, and $(\cdot)^H$ denote the trace, the conjugate, and the conjugate transpose operations, respectively. $|\cdot|$ and $||\cdot||$ stand for the absolute value of a scalar and the Euclidean norm of a vector or matrix, respectively. $\mathbb{E}[\cdot]$ denotes the expectation operation. $\mathbb{C}^{M\times N}$ represents the space of complex-valued matrices.

2.2 System Model and Problem Formulation

In an IRS-aided secure maritime wireless communication system, as shown in Fig. 2.1, the AP at one ship is equipped with N antennas to serve K single-antenna legitimate ships in the presence of M single-antenna eavesdropper ship. An IRS with L reflecting elements assists the secure communication from the AP to the legitimate ships and is equipped with a controller to coordinate with the AP.

For the ease of practical implementation, the IRS is assumed to have the maximal reflection without power loss, and the reflecting elements are designed to maximize the desired signal power to the legitimate ships [13–23]. In addition, unauthorized eavesdropper ships aim to eavesdrop the data streams from the legitimate ships, and the ships are assumed to use the multiple access mechanism to avoid network collision [5, 6, 18–21]. Hence, the IRS chooses the reflecting beamforming to improve the achievable secrecy rate at the legitimate ships and suppress the wiretapped data rate at the eavesdropper ship.

2.2.1 System Model

In the IRS-aided secure maritime wireless communication system, let $\mathscr{K} = \{1, 2, \ldots, K\}$, $\mathscr{M} = \{1, 2, \ldots, M\}$, and $\mathscr{L} = \{1, 2, \ldots, L\}$ denote the legitimate

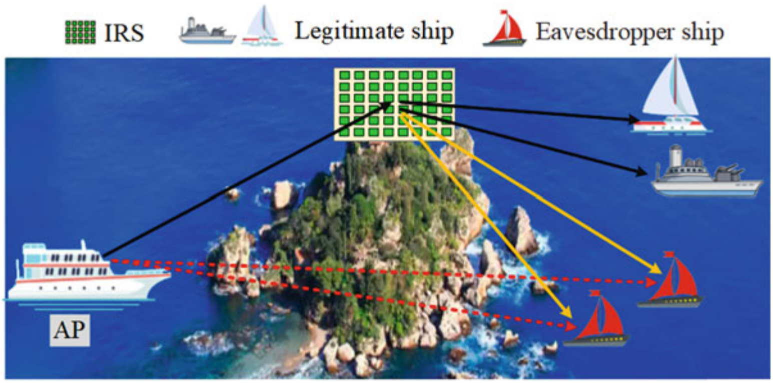


Fig. 2.1 IRS-aided secure maritime wireless communication under multiple eavesdropper ships

ship set, the eavesdropper ship set, and the IRS reflecting element set, respectively. Let $\mathbf{H}_{\text{br}} \in \mathbb{C}^{L\times N}$, $\mathbf{h}_{\text{bu},k}^{H} \in \mathbb{C}^{1\times N}$, $\mathbf{h}_{\text{ru},k}^{H} \in \mathbb{C}^{1\times L}$, $\mathbf{h}_{\text{be},m}^{H} \in \mathbb{C}^{1\times N}$, and $\mathbf{h}_{\text{re},m}^{H} \in \mathbb{C}^{1\times L}$ correspond to the channel coefficients from the AP to the IRS, from the AP to the k-th legitimate ship, from the IRS to the k-th legitimate ship, from the AP to the m-th eavesdropper ship, and from the IRS to the m-th eavesdropper ship, respectively.

Let $\mathbf{G} = \text{diag}(\chi_1 e^{j\theta_1}, \chi_2 e^{j\theta_2}, \ldots, \chi_L e^{j\theta_L})$ denote the reflection coefficient matrix associated with effective phase shifts at the IRS, where $\chi_l \in [0, 1]$ and $\theta_l \in [0, 2\pi]$ represent the amplitude reflection factor and the phase shift coefficient on the combined transmitted signal, respectively. Each phase shift is designed to achieve full reflection, and thus we assume $\chi_l = 1, \forall l \in \mathscr{L}$ in this chapter.

The AP is assumed to use the beamforming vector for the k-th legitimate ship denoted as $\mathbf{v}_k \in \mathbb{C}^{N\times 1}$ and applies the continuous linear precoding [11–16, 23]. Thus, the transmitted signal for all legitimate ships at the AP is given by $\mathbf{x} = \sum_{k=1}^{K} \mathbf{v}_k s_k$, where s_k is the transmitted symbol for the k-th legitimate ship and can be modeled as independent and identically distributed (i.i.d.) random variables with zero mean and unit variance [11–16, 23], and $s_k \sim \mathscr{CN}(0, 1)$. The total transmit power at the AP is subject to the maximum power constraint,

$$\mathbb{E}[\left|\left|\mathbf{x}\right|\right|^2] = \text{Tr}(\mathbf{V}\mathbf{V}^H) \le P_{\max}, \tag{2.1}$$

where $\mathbf{V} \overset{\Delta}{=} [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K] \in \mathbb{C}^{M\times K}$, and $P_{\max}$ is the maximum AP transmit power.

If the AP transmits a secret message to the k-th legitimate ship, the legitimate ship will receive the signal from the AP and the reflected signal from the IRS. Accordingly, the received signal at legitimate ship k can be given by

$$y_k = \underbrace{\left(\mathbf{h}_{\text{ru},k}^{H}\mathbf{G}\mathbf{H}_{\text{br}} + \mathbf{h}_{\text{bu},k}^{H}\right)\mathbf{v}_k s_k}_{\text{desired signal}} + \underbrace{\sum_{i\in\mathscr{K}, i\neq k} \left(\mathbf{h}_{\text{ru},k}^{H}\mathbf{G}\mathbf{H}_{\text{br}} + \mathbf{h}_{\text{bu},k}^{H}\right)\mathbf{v}_i s_i}_{\text{inter}-\text{user interference}} + n_k, \tag{2.2}$$

where n_k denotes the additive complex Gaussian noise (AWGN) with zero mean and variance δ_k^2 at the k-th MU. In (2.2), in addition to the AP signal, each legitimate mobile device also receives inter-user interference (IUI). The signal received by eavesdropper m is given by

$$y_m = \left(\mathbf{h}_{\text{re},m}^{H}\mathbf{G}\mathbf{H}_{\text{br}} + \mathbf{h}_{\text{be},m}^{H}\right)\sum_{k\in\mathscr{K}} \mathbf{v}_k s_k + n_m, \tag{2.3}$$

where n_m is the AWGN of eavesdropper m with variance δ_m^2.

In practical maritime communication systems, neither the AP nor the IRS can obtain perfect CSI [9, 21], due to the transmission and processing delay, as well as the ship mobility. The estimation error and latency of CSI employed in beamforming results in substantial performance loss [21].

Let T denote the delay between the estimated CSI and the real CSI. In other words, upon receiving the pilot sequence from the legitimate ship at the time slot t, the AP uses the channel estimation results in the transmission to the legitimate ships at time slot $t + T$. Hence, the relation between the estimated channel vector $\mathbf{h}(t)$ and the real channel vector $\mathbf{h}(t + T)$ is modeled as

$$\mathbf{h}(t+T) = \rho\mathbf{h}(t) + \sqrt{1-\rho^2}\hat{\mathbf{h}}(t+T), \tag{2.4}$$

where each element in $\hat{\mathbf{h}}(t + T)$ is an independent identically distributed complex Gaussian random variable with N(0, 1). $\rho \in [0, 1]$ as the autocorrelation function (outdated CSI coefficient) of the channel gain $\mathbf{h}(t)$ is given by

$$\rho = J_0(2\pi_{\mathrm{pi}} f_D T), \tag{2.5}$$

where $J_0(\cdot)$ is the zero-th order Bessel function of the first kind. f_D is the Doppler spread, which is generally a function of the velocity (υ) of the transceivers, the carrier frequency (f_c), and the speed of light (c), i.e., $f_D = \upsilon f_c/c$.

As the outdated CSI introduces the channel uncertainty, the actual channel coefficients can be rewritten as

$$\begin{aligned} \mathbf{h}_{\mathrm{bu},k} &= \tilde{\mathbf{h}}_{\mathrm{bu},k} + \Delta\mathbf{h}_{\mathrm{bu},k},\ \forall k \in \mathcal{K} \\ \mathbf{h}_{\mathrm{ru},k} &= \tilde{\mathbf{h}}_{\mathrm{ru},k} + \Delta\mathbf{h}_{\mathrm{ru},k},\ \forall k \in \mathcal{K} \\ \mathbf{h}_{\mathrm{be},m} &= \tilde{\mathbf{h}}_{\mathrm{be},m} + \Delta\mathbf{h}_{\mathrm{be},m},\ \forall m \in \mathcal{M} \\ \mathbf{h}_{\mathrm{re},m} &= \tilde{\mathbf{h}}_{\mathrm{re},m} + \Delta\mathbf{h}_{\mathrm{re},m},\ \forall m \in \mathcal{M}, \end{aligned} \tag{2.6}$$

where $\tilde{\mathbf{h}}_{\mathrm{bu},k}$, $\tilde{\mathbf{h}}_{\mathrm{ru},k}$, $\tilde{\mathbf{h}}_{\mathrm{be},m}$, and $\tilde{\mathbf{h}}_{\mathrm{re},m}$ denote the estimated channel vectors. $\Delta\mathbf{h}_{\mathrm{bu},k}$, $\Delta\mathbf{h}_{\mathrm{ru},k}$, $\Delta\mathbf{h}_{\mathrm{be},m}$, and $\Delta\mathbf{h}_{\mathrm{re},m}$ are the corresponding channel error vectors.

The channel error vectors of each legitimate ship and each eavesdropper are bounded with respect to the Euclidean norm by using norm-bounded error model, i.e.,

$$\left|\left|\Delta\mathbf{h}_i\right|\right|^2 \le (\varsigma_i)^2, i = bu, ru, be, and\ re, \tag{2.7}$$

where ς_{bu}, ς_{ru}, ς_{be}, and ς_{re} refer to the radii of the deterministically bounded error regions.

Based on the channel uncertainty model in Eq. (2.7), the achievable rate of the k-th legitimate ship is given by

$$R_k^{\mathrm{u}} = \log_2\left(1 + \frac{\left|(\mathbf{h}_{\mathrm{ru},k}^H\mathbf{G}\mathbf{H}_{\mathrm{br}} + \mathbf{h}_{\mathrm{bu},k}^H)\mathbf{v}_k\right|^2}{\left|\sum_{i\in\mathcal{K}, i\neq k}(\mathbf{h}_{\mathrm{ru},k}^H\mathbf{G}\mathbf{H}_{\mathrm{br}} + \mathbf{h}_{\mathrm{bu},k}^H)\mathbf{v}_i\right|^2 + \delta_k^2}\right). \tag{2.8}$$

If the m-th eavesdropper attempts to obtain the signal of the k-th legitimate ship, the resulting achievable rate is given by

$$R^{\mathrm{e}}_{m,k} = \log_2\left(1 + \frac{\left|(\mathbf{h}^H_{\mathrm{re},m}\mathbf{G}\mathbf{H}_{\mathrm{br}}+\mathbf{h}^H_{\mathrm{be},m})\mathbf{v}_k\right|^2}{\left|\sum\limits_{i\in\mathscr{K},i\neq k}(\mathbf{h}^H_{\mathrm{re},m}\mathbf{G}\mathbf{H}_{\mathrm{br}}+\mathbf{h}^H_{\mathrm{be},m})\mathbf{v}_i\right|^2+\delta^2_m}\right). \tag{2.9}$$

Since each eavesdropper can eavesdrop any signals from the K legitimate ships, according to [14–25], the achievable secrecy rate at the k-th legitimate ship is given by

$$R^{\mathrm{sec}}_k = \left[R^{\mathrm{u}}_k - \max_{\forall m} R^{\mathrm{e}}_{m,k}\right]^+, \tag{2.10}$$

where $[z]^+ = \max(0, z)$.

2.2.2 Problem Formulation

Our objective is to jointly optimize the AP beamforming matrix $\mathbf{V}$ and the IRS reflecting beamforming matrix $\mathbf{G}$ from the beamforming codebook $\mathscr{F}$ to maximize the worst-case secrecy rate with the data rate constraints, the AP transmit power constraint, and the IRS reflecting unit constraint. As such, the optimization problem is formulated as

$$\begin{aligned}
&\max_{\mathbf{V},\mathbf{G}}\ \min_{\{\Delta\mathbf{h}\}} \sum_{k\in\mathscr{K}} R^{\mathrm{sec}}_k\\
&s.t.\ \ (\mathrm{a}):\ R^{\mathrm{sec}}_k \geq R^{\mathrm{sec,min}}_k,\ \forall k\\
&\qquad (\mathrm{b}):\ (R^{\mathrm{u}}_k) \geq R^{\min}_k,\ \forall k\\
&\qquad (\mathrm{c}):\ \mathrm{Tr}\left(\mathbf{V}\mathbf{V}^H\right) \leq P_{\max}\\
&\qquad (\mathrm{d}):\ |\chi e^{j\theta_l}| = 1,\ 0 \leq \theta_l \leq 2\pi,\ \forall l,
\end{aligned} \tag{2.11}$$

where $R^{\mathrm{sec\,,min}}_k$ is the target secrecy rate of the k-th legitimate ship and $R^{\min}_k$ denotes its target data rate.

The constraints in (2.11a) and (2.11b) represent the worst-case secrecy rate and data rate requirements, respectively. The constraint in (2.11c) is set to satisfy the AP's maximum power constraint. The constraint in (2.11d) is the constraint of the IRS reflecting elements. Obviously, it is challenging to obtain an optimal solution to the optimization (2.11), since the objective function in (2.11) is non-concave with respect to either $\mathbf{V}$ or $\mathbf{G}$, and the coupling of the optimization variables ($\mathbf{V}$ and $\mathbf{G}$) and the unit-norm constraints in (2.11d) are non-convex. In addition, the robust beamforming aims to maximize the worst-case achievable secrecy rate of the system and guarantee the worst-case constraints.

2.3 Problem Transformation Based on RL

The optimization problem in (2.11) is difficult to address as it is non-convex and realistic IRS-aided secure communication systems have time-varying capabilities of legitimate ships, channel quality, and service applications. Moreover, the solution to the problem in (2.11) sometimes converges to a suboptimal solution with greedy-search performance due to the ignorance of the state correlation and the long-term benefit. Hence, it is generally infeasible to apply the traditional optimization techniques, such as AO, SDP, and MM, to achieve an effective secure beamforming policy in maritime communication environments.

Model-free RL is a dynamic programming tool which can be adopted to solve the decision-making problem by learning the optimal solution in dynamic environments [32]. Hence, we model the secure beamforming optimization problem as an RL problem. In RL, the IRS-aided secure communication system is treated as an environment, and the central controller at the AP is a learning agent, with key learning elements defined as follows.

State Space Let $\mathscr{S}$ denote the state space. The state $\boldsymbol{s} \in \mathscr{S}$ includes the channel states of all the ships, the secrecy rate, the transmission data rate of the previous time slot, and the QoS satisfaction level and is defined as

$$\boldsymbol{s} = \left\{ \{\mathbf{h}_k\}_{k\in\mathscr{K}}, \{\mathbf{h}_m\}_{m\in\mathscr{M}}, \{R_k^{\mathrm{sec}}\}_{k\in\mathscr{K}}, \{R_k\}_{k\in\mathscr{K}},\ \{\mathrm{QoS}_k\}_{k\in\mathscr{K}} \right\}, \tag{2.12}$$

where $\mathbf{h}_k$ and $\mathbf{h}_m$ are the channel coefficients of the k-th legitimate ship and m-th eavesdropper, respectively. QoS_k is the feedback QoS satisfaction level of the k-th MU, consisting of both the minimum secrecy rate satisfaction level in (2.11a) and the minimum data rate satisfaction level in (2.11b).

Action Space Let $\mathscr{A}$ denote the action space. According to the observed state $\boldsymbol{s}$, the central controller chooses the beamforming vector $\{\mathbf{v}_k\}_{k\in\mathscr{K}}$ at the AP and the IRS reflecting beamforming coefficient or phase shift $\{\theta_l\}_{l\in\mathscr{L}}$ at the IRS with the action $\boldsymbol{a} \in \mathscr{A}$ given by

$$\boldsymbol{a} = \left\{ \{\mathbf{v}_k\}_{k\in\mathscr{K}}, \{\theta_l\}_{l\in\mathscr{L}} \right\}. \tag{2.13}$$

Transition Probability Let $\mathscr{T}(s'|s, a)$ be transition probability that represents the probability of transitioning to a new state $s' \in \mathscr{S}$, given the action $\mathbf{a}$ executed in the state $\mathbf{s}$.

Reward Function In RL, the reward acts as a signal for the controller, the learning agent, to evaluate the performance of the secure beamforming policy at the current state. The system performance will be enhanced if the reward function at each learning step correlates with the desired objective. Thus, it is important to design an efficient reward function to improve the legitimate ships' QoS satisfaction levels.

The reward function that represents the optimization objective consists of the system secrecy rate of all legitimate ships and the level to guarantee the QoS requirements. Thus, the QoS-aware reward function is given by

$$r = \underbrace{\sum_{k \in \mathscr{K}} R_k^{\text{sec}}}_{\text{part 1}} - \underbrace{\sum_{k \in \mathscr{K}} \mu_1 p_k^{\text{sec}}}_{\text{part 2}} - \underbrace{\sum_{k \in \mathscr{K}} \mu_2 p_k^{\text{u}}}_{\text{part 3}}, \tag{2.14}$$

where

$$p_k^{\text{sec}} = \begin{cases} 1, & \text{if } R_k^{\text{sec}} < R_k^{\text{sec},\min}, \forall k \in \mathscr{K} \\ 0, & \text{otherwise}, \end{cases} \tag{2.15}$$

$$p_k^{\text{u}} = \begin{cases} 1, & \text{if } R_k < R_k^{\min}, \forall k \in \mathscr{K} \\ 0, & \text{otherwise.} \end{cases} \tag{2.16}$$

In (2.14), the first term represents the immediate utility (system secrecy rate), and the second and third terms are the cost functions in terms of the unsatisfied secrecy rate requirement and the unsatisfied minimum rate requirement, respectively. The coefficients μ_1 and μ_2 are the positive constants of the latter two terms, respectively, as a trade-off between the utility and cost [33–35].

The goals of (2.15) and (2.16) impose the QoS satisfaction levels of both the secrecy rate and the minimum data rate requirements, respectively. If the QoS requirement is satisfied in the current time slot, then $p_k^{\text{sec}} = 0$ or $p_k^{\text{u}} = 0$, indicating no punishment of the reward function due to the successful QoS guarantees.

The learning agent aims to search for an optimal policy π^* (π is a mapping from states in $\mathscr{S}$ to the probabilities of choosing an action in $\mathscr{A}$: $\pi(s) : \mathscr{S} \to \mathscr{A}$) that maximizes the long-term expected discounted reward. The cumulative discounted reward function can be defined as

$$U_t = \sum_{\tau=0}^{\infty} \gamma_\tau r_{t+\tau+1}, \tag{2.17}$$

where $\gamma \in (0, 1]$ denotes the discount factor. Under a certain policy π, the state-action function of the agent with a state-action pair (s, a) is given by

$$Q^\pi(s_t, a_t) = \mathbb{E}_\pi \left[U_t | s_t = s, a_t = a \right]. \tag{2.18}$$

The Q-table is updated based on the Bellman equation as follows:

$$\begin{aligned} Q^\pi(s_t, a_t) = \mathbb{E}_\pi \Bigg[& r_t + \gamma \sum_{s_{t+1} \in \mathscr{S}} \mathscr{T}(s_{t+1} | s_t, a_t) \\ & \sum_{a_{t+1} \in \mathscr{A}} \pi(s_{t+1}, a_{t+1}) Q^\pi(s_{t+1}, a_{t+1}) \Bigg]. \end{aligned} \tag{2.19}$$

The optimal action-value function in (2.17) is equivalent to the Bellman optimality equation given by

$$Q^*(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}), \tag{2.20}$$

and the state-value function is achieved as follows:

$$V(s_t) = \max_{a_t \in \mathscr{A}} Q(s_t, a_t). \tag{2.21}$$

In addition, the Q-value is updated as follows:

$$Q_{t+1}(s_t, a_t) = (1 - \alpha_t) Q_t(s_t, a_t) + \alpha_t \left[r_t + \gamma V_t(s_{t+1}) \right], \tag{2.22}$$

where $\alpha_t \in (0, 1]$ is the learning rate. Based on a lookup Q-table $Q(s, a)$, the agent selects actions following the greedy policy in each learning step [32]. More specifically, the action with the maximum Q-table value is chosen with probability $1 - \varepsilon$, whereas a random action is picked with probability ε to avoid achieving stuck at nonoptimal policies [32]. Once the optimal Q-function $Q^*(s, a)$ is achieved, the optimal policy is given by

$$\pi^*(s, a) = \arg \max_{a \in \mathscr{A}} Q^*(s, a). \tag{2.23}$$

2.4 Deep PDS-PER Learning-Based Secure Beamforming

The secure beamforming policy discussed in Sect. 2.3 can be numerically achieved by using Q-Learning, policy gradient, and deep Q-network (DQN) algorithms [32]. However, Q-learning is not an efficient learning algorithm because it cannot deal with continuous state space and it has slow learning convergence speed. The policy gradient algorithm has the ability to handle continuous state-action spaces, but it may converge to a suboptimal solution. In addition, it is intractable for Q-learning and policy gradient algorithms to solve the optimization problem under high-dimensional input state space. Although DQN performs well in policy learning under high-dimensional state space, its nonlinear Q-function estimator may lead to unstable learning process.

Considering the fact that the IRS-aided secure communication system has high-dimensional and high-dynamical characteristics according to the system state that is defined in (2.12) and uncertain CSI that is shown in (2.4) and (2.6), we propose a deep PDS-PER learning-based secure beamforming approach, as shown in Fig. 2.2, where PDS learning and PER mechanisms are utilized to enable the learning agent to learn and adapt faster in dynamic environments. In detail, the agent utilizes the observed state (i.e., CSI, previous secrecy rate, QoS satisfaction level) and the feedback reward from the environment as well as the historical experience from the

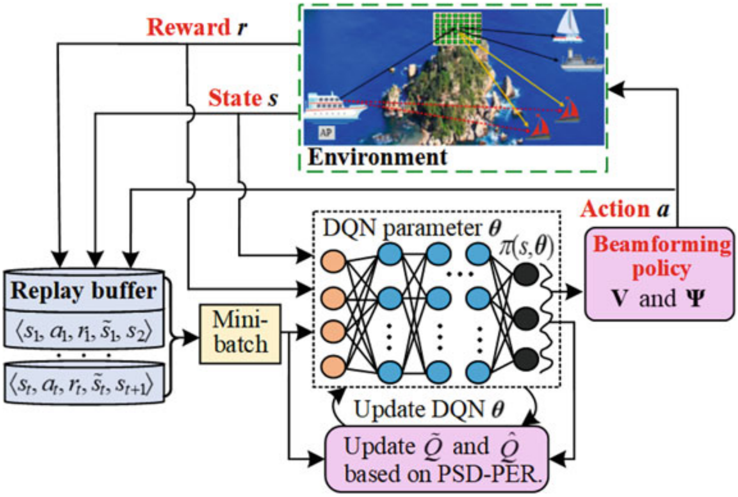

Reward r
State s
AP
Environment
Action a
DQN parameter θ
$\pi(s,\theta)$
Beamforming policy
$\mathbf{V}$ and $\mathbf{\Psi}$
Replay buffer
$\langle s_1, a_1, r_1, \tilde{s}_1, s_2 \rangle$
$\langle s_t, a_t, r_t, \tilde{s}_t, s_{t+1} \rangle$
Mini-batch
Update DQN θ
Update $\tilde{Q}$ and $\hat{Q}$ based on PSD-PER

Fig. 2.2 Deep PDS-PER learning-based beamforming for IRS-aided secure communications

replay buffer to train its learning model. After that, the agent employs the trained model to make decision (beamforming matrices **V** and **G**) based on its learned policy. The procedures of the proposed learning-based secure beamforming are provided in the following subchapters.

Note that the policy optimization (in terms of the AP's beamforming matrix **V** and the RIS's reflecting beamforming matrix **G**) in the IRS-aided secure communication system can be performed at the AP and that the optimized reflecting beamforming matrix can be transferred in an offline manner to the IRS by the controller to adjust the corresponding reflecting elements accordingly.

2.4.1 Proposed Deep PDS-PER Learning

As discussed in Chap. 2, CSI is unlikely to be known accurately due to the transmission delay, processing delay, and mobility of ships. At the same time, beamforming with outdated CSI will decrease the secrecy capacity, and therefore, a fast optimization solution needs to be designed to reduce processing delay. PDS learning as a well-known algorithm has been used to improve the learning speed by exploiting extra partial information (e.g., the previous location information and the mobility velocity of legitimate ships or eavesdropper ship that affect the channel coefficients) and search for an optimized policy in dynamic environments [33–35]. Motivated by this, we devise a modified deep PDS learning to trace the environment dynamic characteristics and then adjust the transmit beamforming at the AP and the reflecting elements at the IRS accordingly, which can speed up the learning efficiency in dynamic environments.

PDS learning can be defined as an immediate system state $\tilde{s}_t \in \mathscr{S}$ that happens after executing an action a_t at the current state s_t and before the next time state s_{t+1}. In detail, the PDS learning agent takes an action a_t at state s_t and then will receive a known reward $r^{\mathrm{k}}(s_t, a_t)$ from the environment before transitioning the current state s_t to the PDS state $\tilde{s}_t$ with a known transition probability $\mathscr{T}^{\mathrm{k}}(\tilde{s}_t|s_t, a_t)$. After that, the PDS state further transforms to the next state s_{t+1} with an unknown transition probability $\mathscr{T}^{\mathrm{u}}(s_{t+1}|\tilde{s}_t, a_t)$ and an unknown reward $r^{\mathrm{u}}(s_t, a_t)$, which corresponds to the wireless CSI dynamics. In PDS learning, s_{t+1} is independent of s_t given the PDS state $\tilde{s}_t$, and the reward $r(s_t, a_t)$ is decomposed into the sum of $r^{\mathrm{k}}(s_t, a_t)$ and $r^{\mathrm{u}}(s_t, a_t)$ at $\tilde{s}_t$ and s_{t+1}, respectively. Mathematically, the state transition probability in PDS learning from s_t to s_{t+1} admits

$$\mathscr{T}(s_{t+1}|s_t, a_t) = \sum\nolimits_{\tilde{s}_t} \mathscr{T}^{\mathrm{u}}(s_{t+1}|\tilde{s}_t, a_t)\mathscr{T}^{\mathrm{k}}(\tilde{s}_t|s_t, a_t). \tag{2.24}$$

Moreover, it can be verified that the reward of the current state-action pair (s_t, a_t) is expressed by

$$r(s_t, a_t) = r^{\mathrm{k}}(s_t, a_t) + \sum\nolimits_{\tilde{s}_t} \mathscr{T}^{\mathrm{k}}(\tilde{s}_t|s_t, a_t)r^{\mathrm{u}}(\tilde{s}_t, a_t). \tag{2.25}$$

At the time slot t, the PDS action-value function $\tilde{Q}(\tilde{s}_t, a_t)$ of the current PDS state-action pair $(\tilde{s}_t, a_t)$ is defined as

$$\tilde{Q}(\tilde{s}_t, a_t) = r^{\mathrm{u}}(\tilde{s}_t, a_t) + \gamma \sum_{s_{t+1}} \mathscr{T}^{\mathrm{u}}(s_{t+1}|\tilde{s}_t, a_t)V(s_{t+1}). \tag{2.26}$$

By employing the extra information (the known transition probability $\mathscr{T}^{\mathrm{k}}(\tilde{s}_t|s_t, a_t)$ and known reward $r^{\mathrm{k}}(s_t, a_t)$), the Q-function $\hat{Q}(s_t, a_t)$ in PDS learning can be further expanded under all state-action pairs (s, a), which is expressed by

$$\hat{Q}(s_t, a_t) = r^{\mathrm{k}}(s_t, a_t) + \sum\nolimits_{\tilde{s}_t} \mathscr{T}^{\mathrm{k}}(\tilde{s}_t|s_t, a_t)\tilde{Q}(\tilde{s}_t, a_t). \tag{2.27}$$

The state-value function in PDS learning is defined by

$$\hat{V}_t(s_t) = \sum\nolimits_{s_{t+1}} \mathscr{T}^{\mathrm{k}}(s_{t+1}|s_t, a_t)\tilde{V}(s_{t+1}), \tag{2.28}$$

where $\tilde{V}_t(s_{t+1}) = \max\limits_{a_t \in \mathscr{A}} \tilde{Q}_t(\tilde{s}_{t+1}, a_t)$. At each time slot, the PDS action-value function $\tilde{Q}(\tilde{s}_t, a_t)$ is updated by

$$\tilde{Q}_{t+1}(\tilde{s}_t, a_t) = (1 - \alpha_t)\tilde{Q}_t(\tilde{s}_t, a_t) + \alpha_t \left(r^{\mathrm{u}}(\tilde{s}_t, a_t) + \gamma \hat{V}_t(s_{t+1})\right). \tag{2.29}$$

After updating $\tilde{Q}_{t+1}(\tilde{s}_t, a_t)$, the action-value function $\hat{Q}_{t+1}(s_t, a_t)$ can be updated by plugging $\tilde{Q}_{t+1}(\tilde{s}_t, a_t)$ into (2.27).

After presenting in the above-modified PDS learning, a deep PDS learning algorithm is presented. In the presented learning algorithm, the traditional DQN is adopted to estimate the action-value Q-function $Q(s, a)$ by using $Q(s, a; \theta)$, where θ denotes the DNN parameter. The objective of DQN is to minimize the following loss function at each time slot:

$$\begin{aligned}\mathscr{L}(\theta_t) = \left[\left(\hat{V}_t(s_t; \theta_t) - \hat{Q}(s_t, a_t; \theta_t)\right)^2\right] = \Big[\Big(r(s_t, a_t)\\ +\gamma \max_{a_{t+1}\in\mathscr{A}} \hat{Q}_t(s_{t+1}, a_{t+1}; \theta_t) - \hat{Q}(s_t, a_t; \theta_t)\Big)^2\Big],\end{aligned} \tag{2.30}$$

where $\hat{V}_t(s_t; \theta_t) = r(s_t, a_t) + \gamma \max_{a_{t+1}\in\mathscr{A}} \hat{Q}_t(s_{t+1}, a_{t+1}; \theta_t)$ is the target value. The error between $\hat{V}_t(s_t; \theta_t)$ and the estimated value $\hat{Q}(s_t, a_t; \theta_t)$ is usually called temporal difference (TD) error, which is expressed by

$$\delta_t = \hat{V}_t(s_t; \theta_t) - \hat{Q}(s_t, a_t; \theta_t). \tag{2.31}$$

The DNN parameter θ is achieved by taking the partial differentiation of the objective function (2.30) with respect to θ, which is given by

$$\theta_{t+1} = \theta_t + \beta\nabla\mathscr{L}(\theta_t), \tag{2.32}$$

where β is the learning rate of θ and $\nabla(\cdot)$ denotes the first-order partial derivative.

Accordingly, the policy $\hat{\pi}_t(s)$ of the modified deep PDS learning algorithm is given by

$$\hat{\pi}_t(s) = \arg\max_{a_t\in\mathscr{A}} \hat{Q}(s_t, a_t; \theta_t). \tag{2.33}$$

Although DQN is capable of performing well in policy learning with continuous and high-dimensional state space, DNN may learn ineffectively and cause divergence owing to the nonstationary targets and correlations between samples. Experience replay is utilized to avoid the divergence of the RL algorithm. However, classical DQN uniformly samples each transition $e_t = \langle s_t, a_t, r_t, \tilde{s}_t, s_{t+1}\rangle$ from the experience replay, which may lead to an uncertain or negative effect on learning a better policy. The reason is that different transitions (experience information) in the replay buffer have different importance for the learning policy, and sampling every transition equally may unavoidably result in inefficient usage of meaningful transitions. Therefore, a prioritized experience replay (PER) scheme has been presented to address this issue and enhance the sampling efficiency [36, 37], where the priority of transition is determined by the values of TD error. In PER, a transition with higher absolute TD error has higher priority in the sense that it has more aggressive correction for the action-value function.

In the deep PDS-PER learning algorithm, similar to classical DQN, the agent collects and stores each experience $e_t = \langle s_t, a_t, r_t, \tilde{s}_t, s_{t+1} \rangle$ into its experience replay buffer, and DNN updates the parameter by sampling a minibatch of tuples from the replay buffer. So far, PER was adopted only for DRL and Q-learning and has never been employed with the PDS learning algorithm to learn the dynamic information. In this chapter, we further extend this PER scheme to enable prioritized experience replay in the proposed deep PDS-PER learning framework, in order to improve the learning convergence rate.

The probability of sampling transition i (experience i) based on the absolute TD error is defined by

$$p(i) = \left|\delta(i)\right|^{\eta_1} \Big/ \sum\nolimits_{j'} \left|\delta(j')\right|^{\eta_1}, \tag{2.34}$$

where the exponent η_1 weights how legitimate shipch prioritization is used, with $\eta_1 = 0$ corresponding to being uniform sampling. The transition with higher $p(i)$ will be more likely to be replayed from the replay buffer, which is associated with very successful attempts by preventing the DNN from being over-fitting. With the help of PER, the proposed deep PDS-PER learning algorithm tends to replay valuable experience and hence learns more effectively to find the best policy.

It is worth noting that experiences with high absolute TD error are more frequently replayed, which alters the visitation frequency of some experiences and hence causes the training process of the DNN prone to diverge. To address this problem, importance-sampling (IS) weights are adopted in the calculation of weight changes:

$$W(i) = \big(D \cdot p(i)\big)^{-\eta_2}, \tag{2.35}$$

where D is the size of the experience replay buffer and the parameter η_2 is used to adjust the amount of correction used.

Accordingly, by using the PER scheme into the deep PDS-PER learning, the DNN loss function (2.30) and the corresponding parameters are rewritten respectively as follows:

$$\mathscr{L}(\theta_t) = \frac{1}{H} \sum_{i=1}^{H} \big(W_i \mathscr{L}_i(\theta_t)\big), \tag{2.36}$$

$$\theta_{t+1} = \theta_t + \beta \delta_t \nabla_\theta \mathscr{L}(\theta_t). \tag{2.37}$$

The presented deep PDS-PER learning can converge to the optimal $\hat{Q}(s_t, a_t)$ of the MDP with probability 1 when the learning rate sequence α_t meets the following conditions: $\alpha_t \in [0, 1)$, $\sum_{t=0}^{\infty} \alpha_t = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$, where the aforementioned requirements have been appeared in most of the RL algorithms and they are not specific to the proposed deep PDS-PER learning algorithm. The existing references [34] and [35] have provided the proof.

2.4.2 Secure Beamforming Based on Proposed Deep PDS-PER Learning

Similar to most DRL algorithms, our proposed deep PDS-PER learning-based secure beamforming approach consists of two stages, i.e., the training stage and implementation stage. The training process of the proposed approach is shown in **Algorithm 1**. A central controller at the AP is responsible for collecting environment information and making decision for secure beamforming.

In the training stage, similar to RL-based policy control, the controller initializes network parameters and observes the current system state including CSI of all ships, the previous predicted secrecy rate, and the transmission data rate. Then, the state vector is input into DQN to train the learning model. The ε-greedy scheme is leveraged to balance both the exploration and exploitation, i.e., the action with the maximum reward is selected probability 1-ε according to the current information (exploitation, which is known knowledge), while a random action is chosen with probability ε based on the unknown knowledge (i.e., keep trying new actions, hoping it brings even higher reward (exploration, which is unknown knowledge)).

After executing the selected action, the agent receives a reward from the environment and observes the state transition from s_t to PDS state $\tilde{s}_t$ and then to the next state s_{t+1}. Then, PDS learning is used to update the PDS action-value function $\tilde{Q}(\tilde{s}_t, a_t; \theta_t)$ and Q-function $\hat{Q}(s_t, a_t; \theta_t)$, before collecting and storing the transition tuple (also called experience) $e_t = \langle s_t, a_t, r_t, \tilde{s}_t, s_{t+1} \rangle$ into the experience replay memory buffer $\mathscr{D}$, which includes the current system state, selected action, instantaneous reward, and PDS state along with the next state. The experience in the replay buffer is selected by the PER scheme to generate minibatches and they are used to train DQN.

In detail, the priority of each transition $p(i)$ is calculated by using (34) and then get its IS weight $W(i)$ in (35), where the priorities ensure that high-TD-value ($\delta(i)$) transitions are replayed more frequently. The weight $W(i)$ is integrated into deep PDS learning to update both the loss function $\mathscr{L}(\theta)$ and DNN parameter θ. Once DQN converges, the deep PDS-PER learning model is achieved.

After adequate training in **Algorithm 1**, the learning model is loaded for the implementation stage. During the implementation stage, the controller uses the trained learning model to output its selected action a by going through the DNN parameter θ, with the observed state s from the IRS-aided secure communication system. Specifically, it chooses an action a, with the maximum value based on the trained deep PDS-PER learning model. Afterward, the environment feeds back an instantaneous reward and a new system state to the agent. Finally, the beamforming

Algorithm 1 Deep PDS-PER learning-based secure beamforming

```
Input: IRS-aided secure communication simulator and QoS requirements of all legitimate ships (e.g., minimum secrecy rate and transmission rate).
Initialize: DQN with initial Q-function $Q(s, a; \theta)$, parameters $\theta$, learning rate $\alpha$ and $\beta$.
Initialize: experience replay buffer $\mathscr{D}$ with size $D$, and minibatch size $H$.
for each episode =1, 2, ..., $N^{\text{epi}}$ do
  Observe an initial system state $s$;
  for each time step $t$=0, 1, 2, ..., $T$ do
    Select action based on the $\varepsilon$-greedy policy at current state $s_t$: choose a random action $a_t$ with probability $\varepsilon$;
    Otherwise, $a_t = \arg\max_{a_t \in \mathscr{A}} Q(s_t, a_t; \theta_t)$;
    Execute action $a_t$, receive an immediate reward $r^{\text{k}}(s_t, a_t)$ and observe the state transition from $s_t$ to PDS state $\tilde{s}_t$ and then to the next state $s_{t+1}$;
    Update the reward function $r(s_t, a_t)$ under PDS learning using (2.25);
    Update the PDS action-value function $\tilde{Q}(\tilde{s}_t, a_t; \theta_t)$ using (2.29);
    Update the Q-function $\hat{Q}(s_t, a_t; \theta_t)$ using (2.25);
    Store PDS experience $e_t = \langle s_t, a_t, r_t, \tilde{s}_t, s_{t+1} \rangle$ in experience replay buffer $\mathscr{D}$, if $\mathscr{D}$ is full, remove least used experience from $\mathscr{D}$;
    for $i$= 1, 2, ..., $H$ do
      Sample transition $i$ with the probability $p(i)$ using (2.34);
      Calculate the absolute TD-error $|\delta(i)|$ in (2.31);
      Update the corresponding IS weight $W_i$ using (2.35);
      Update the priority of transition $i$ based on $|\delta(i)|$;

    end for
    Update the loss function $\mathscr{L}(\theta)$ and parameter $\theta$ of DQN using (2.36) and (2.37), respectively;
  end for
end for
Output: Return the deep PDS-PER learning model.
```

matrix $\mathbf{V}^*$ at the AP and the phase shift matrix $\mathbf{G}^*$ (reflecting beamforming) at the IRS are achieved according to the selected action.

We would like to point out that the training stage needs a powerful computation server which can be performed offline at the AP, while the implementation stage can be completed online. The trained learning model requires to be updated only when the environment (IRS-aided secure communication system) has experienced great changes, mainly depending on the environment dynamics and service requirements.

2.4.3 Computational Complexity Analysis

For the training stage, in DNN, let L, Z_0, and Z_l denote the training layers, the size of the input layer (which is proportional to the number of states), and the number of neurons in the l-th layer, respectively. The computational complexity in each time step for the agent is $O(Z_0 Z_l + \sum_{l=1}^{L-1} Z_l Z_{l+1})$. In the training phase, each minibatch has N^{epi} episodes with each episode being T time steps, and each trained model is completed iteratively until convergence. Hence, the total computational complexity in DNN is $O\left(N^{\text{epi}}T(Z_0 Z_l + \sum_{l=1}^{L-1} Z_l Z_{l+1})\right)$. The high computational complexity of the DNN training phase can be performed offline for a finite number of episodes at a centralized powerful unit (such as the AP).

In our proposed deep PDS-PER learning algorithm, PDS learning and PER schemes are utilized to improve the learning efficiency and enhance the convergence speed, which requires extra computational complexity. In PDS learning leaning, since the set of PDS states is the same as the set of MDP states $\mathscr{S}$ [30–32], the computational complexity of the classical DQN algorithm and the deep PDS learning algorithm are $O(|\mathscr{S}|^2 \times |\mathscr{A}|)$ and $O(2|\mathscr{S}|^2 \times |\mathscr{A}|)$, respectively. In PER, since the relay buffer size is D, the system requires to make both updating and sampling $O\left(\log_2 D\right)$ operations, so the computational complexity of the PER scheme is $O\left(\log_2 D\right)$.

According the above analysis, the complexity of the classical DQN algorithm is $O\left(IN^{\text{epi}}T(Z_0 Z_l + \sum_{l=1}^{L-1} Z_l Z_{l+1}) + |\mathscr{S}|^2 \times |\mathscr{A}|\right)$, while the proposed deep PDS-PER learning algorithm is $O\left(IN^{\text{epi}}T(Z_0 Z_l + \sum_{l=1}^{L-1} Z_l Z_{l+1}) + 2|\mathscr{S}|^2 \times |\mathscr{A}| + \log_2 D\right)$, indicating that the complexity of the proposed algorithm is slightly higher than the classical DQN learning algorithm. However, our proposed algorithm achieves better performance than that of the classical DQN algorithm, which will be shown in the next section.

2.4.4 Implementation Details of DRL

This subchapter provides extensive details regarding the generation of training, validation, and testing dataset production.

Generation of Training As shown in Fig. 2.3, K single-antenna legitimate ships and M single-antenna eavesdropper ship are randomly located in the 100m × 100m half right-hand side rectangular grid plane of Fig. 2.3 (light blue area) in a two-dimensional x-y rectangular grid plane. The AP and the IRS are located at (0, 0) and (100, 100) in meter (m), respectively. The x-y grid has dimensions 100m × 100m with a resolution of 2.5m, i.e., a total of 1600 points.

In the presented IRS-assisted maritime wireless communication system, the system beamforming codebook $\mathscr{F}$ includes the AP's beamforming codebook $\mathscr{F}_{\text{BS}}$ and the IRS's beamforming codebook $\mathscr{F}_{\text{IRS}}$. Both the AP's beamforming matrix $\mathbf{V}$ and the IRS's reflecting beamforming matrix $\mathbf{G}$ are picked from the pre-defined codebook $\mathscr{F}_{\text{BS}}$ and $\mathscr{F}_{\text{IRS}}$, respectively. The data points of sampled channel vector and the corresponding reward vector $\langle \mathbf{h}, \mathbf{r} \rangle$ are added into the DNN training dataset $\mathscr{D}$. The sampled channel, $\mathbf{h}$, is the input to DQN. All the samples are normalized by using the normalization scheme to realize a simple per-dataset scaling. After training, the selected AP's beamforming matrix $\mathbf{V}$ and IRS's beamforming matrix

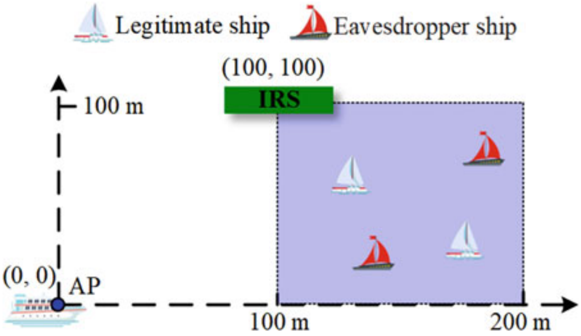


Fig. 2.3 Simulation setup

G with the highest achievable reward are used to reflect the security communication performance.

The DQN learning model is trained using an empirical hyper-parameter, where DNN is trained for 1000 epochs with 128 minibatches being utilized in each epoch. In the training process, 80% and 20% of all generated data are selected as the training and validation (test) datasets, respectively. The experience replay buffer size is 32,000 where the corresponding samples are randomly sampled from this number of the most recent experiences.

DQN Structure The DQN model is designed as a multilayer perceptron network, which is also referred to as the feedforward fully connected network. Note here that multilayer perceptron network is widely used to build an advanced estimator, which fulfills the relation between the environment descriptors and the beamforming matrices (both the AP's beamforming matrix and the IRS's reflecting beamforming matrix).

The DQN model is comprised of L layers, as illustrated in Fig. 2.2, where the first layer is the input layer, the last layer is the output layer, and the remaining layers are the hidden layers. The l-th hidden layer in the network has a stack of neurons, each of which connects all the outputs of the previous layer. Each unit operates on a single input value outputting another single value. The input of the input layer consist of the system states, i.e., channel samples, the achievable rate, and QoS satisfaction level information in the last time slot, while the output layer outputs the predicted reward values with beamforming matrices in terms of the AP's beamforming matrix and the IRS's reflecting beamforming matrix. The DQN construction is used for training stability. The network parameters will be provided in the next chapter.

Training Loss Function The objective of DRL model is to find the best beamforming matrices, i.e., **V** and **G**, from the beamforming codebook with the highest achievable reward from the environment. In this case, having the highest achievable reward estimation, the regression loss function is adopted to train the learning model, where DNN is trained to make its output, $\hat{\mathbf{r}}$, as close as possible to the desired normalized reward, $\bar{\mathbf{r}}$. Formally, the training is driven by minimizing the loss function, $\mathscr{L}(\theta)$, defined as

$$\mathscr{L}(\theta) = MSE\left(\hat{\mathbf{r}}, \bar{\mathbf{r}}\right), \tag{2.38}$$

where θ is the set of all DNN parameters and $MSE(\cdot)$ denotes the mean squared error between $\hat{\mathbf{r}}$ and $\bar{\mathbf{r}}$. Note that the outputs of DNN, $\hat{\mathbf{r}}$ can be acted as functions of θ and the inputs of DNN are the system states shown in (12) in the chapter.

2.5 Simulation Results and Analysis

We evaluate the performance of the IRS-aided secure maritime wireless communication system. The background noise power of legitimate ships and eavesdropper ships is equal to -90 dBm. We set the number of antennas at the AP as $N = 4$, the number of legitimate ships as $K = 2$, and the number of eavesdropper ships as $M = 2$. The transmit power $P_{\max}$ at the AP varies between 15 dBm and 40 dBm, the number of IRS elements L varies between 10 and 60, and the outdated CSI coefficient ρ varies from 0.5 to 1 for different simulation settings. The minimum secrecy rate and the minimum transmission data rate are 3 bits/s/Hz and 5 bits/s/Hz, respectively.

The path loss model is defined by $PL = (PL_0 - 10\varsigma \log 10(d/d_0))$ dB, where $PL_0 = 30$ dB is the path loss at the reference distance $d_0 = 1$ m [9, 38], $\varsigma = 3$ is the path loss exponent, and d is the distance from the transmitter to the receiver. The learning model consists of three connected hidden layers, containing 500, 250, and 200 neurons [39], respectively. The learning rate is set to $\alpha = 0.002$ and the discount factor is set to $\gamma = 0.95$. The exploration rate ε is linearly annealed from 0.8 to 0.1 over the beginning 300 episodes and remains constant afterward. The parameters μ_1 and μ_2 in (12) are set to $\mu_1 = \mu_2 = 2$ to balance the utility and cost [33–35]. Similar to the IRS-aided communication systems [9, 13] and [17], the path loss exponent from the AP to the ships is set to 3.2, from the AP to IRS is set to 2.2, and from the IRS to the ships is set to 2.2.

The selection of the network parameters decides the learning convergence speed and efficiency. Here, we take the network parameters, i.e., the learning rate, as an example to demonstrate the importance of the network parameter selection. Figure 2.4 shows the average system reward versus training episodes under the learning rate, i.e., $\alpha = 0.002$. It can be observed that two learning algorithms have different convergence speed and reward performances. Specifically, there exist oscillations in behavior for the reward performance before achieving the convergence. We can see that the proposed deep PSD-PER algorithm obtains better convergence speed and reward value than those of the DQN algorithm.

In addition, simulation results are provided to evaluate the performance of the proposed deep PDS-PER learning-based secure beamforming approach (denoted as deep PDS-PER beamforming) in the IRS-aided secure communication system and compare the proposed approach with the following existing approaches:

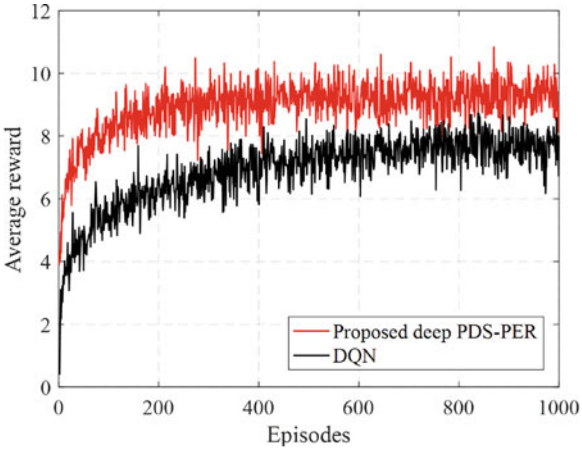


Fig. 2.4 Average reward performance versus episodes

- The classical DQN-based secure beamforming approach (denoted as DQN-based beamforming), where DNN is employed to estimate the Q-value function, when acting and choosing the secure beamforming policy corresponding to the highest Q-value.
- The optimal AP's transmit beamforming approach without IRS assistance (denoted as optimal AP without IRS). Without IRS, the optimization problem (2.11) transformed as

$$\begin{aligned} &\max_{\mathbf{V}} \min_{\{\Delta\mathbf{h}\}} \sum_{k\in\mathscr{K}} R_k^{\text{sec}} \\ s.t. \quad & (\text{a}):\ R_k^{\text{sec}} \geq R_{k,\min}^{\text{sec}},\ \forall k \\ & (\text{b}):\ (R_k^{\text{u}}) \geq R_k^{\min},\ \forall k \\ & (\text{c}):\ \operatorname{Tr}\left(\mathbf{V}\mathbf{V}^H\right) \leq P_{\max} \\ & (\text{d}):\ |\chi e^{j\theta_l}| = 1,\ 0 \leq \theta_l \leq 2\pi,\ \forall l. \end{aligned} \tag{2.39}$$

From the optimization problem (2.39), the system only needs to optimize the AP's transmit beamforming matrix . Problem (2.39) is non-convex due to the rate constraints, and hence we consider semidefinite programming (SDP) relaxation to solve it. After transforming problem (2.39), into a convex optimization problem, we can use CVX to obtain the solution [12–16].

Figure 2.5 shows the average secrecy rate versus the maximum transmit power $P_{\max}$, when $L = 40$ and $\rho = 0.95$. As expected, the secrecy rates of all the approaches enhance monotonically with increasing $P_{\max}$. The reason is that when $P_{\max}$ increases, the received SINR at legitimate ships improves, leading to the performance improvement. In addition, we find that our proposed learning approach outperforms the Baseline1 approach. In fact, our approach jointly optimizes the beamforming matrices **V** and **G**, which can simultaneously facilitate more favorable channel propagation benefit for legitimate ships and impair eavesdropper ship, while

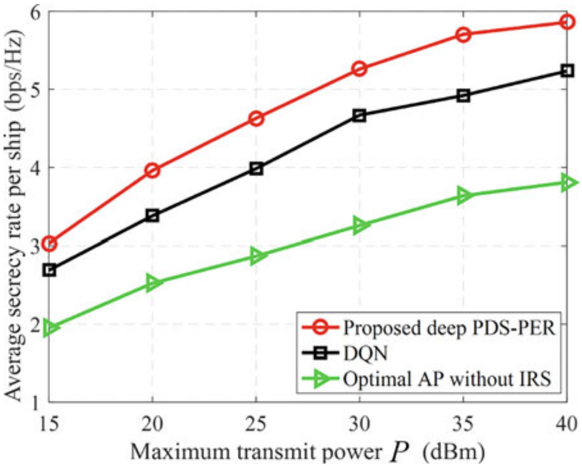


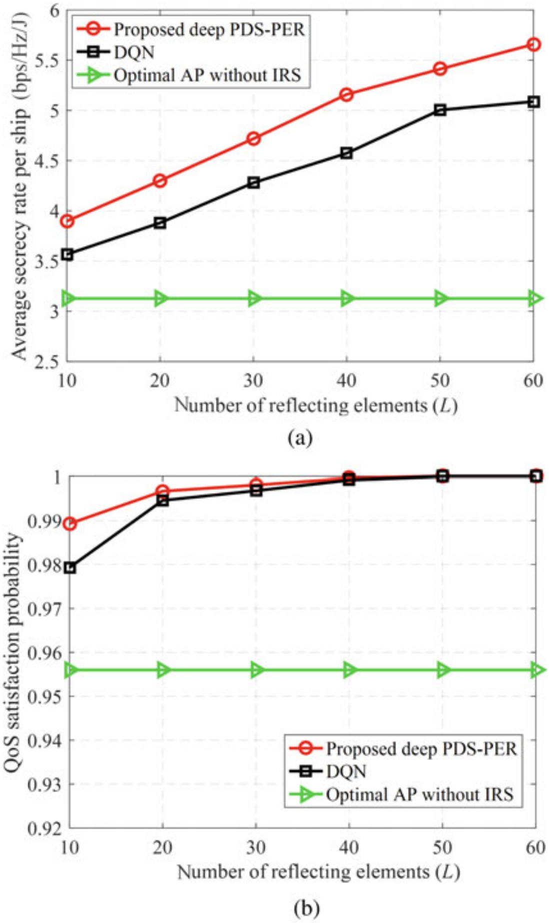
Fig. 2.5 Performance comparisons versus the maximum transmit power at the AP

the Baseline1 approach optimizes the beamforming matrixes in an iterative way. Moreover, our proposed approach has higher performance than DQN in terms of secrecy rate, due to its efficient learning capacity by utilizing PDS learning and PER schemes in the dynamic environment. From Fig. 2.5, we also find that the three IRS-assisted secure beamforming approaches provide significant higher secrecy rate probability than the traditional system without IRS. This indicates that the IRS can effectively guarantee secure communication via reflecting beamforming, where reflecting elements (IRS-induced phases) at the IRS can be adjusted to maximize the received SINR at legitimate ships and suppress the wiretapped rate at eavesdropper ships.

In Fig. 2.6, the achievable secrecy rate and QoS satisfaction level performance of all approaches are evaluated through changing the IRS elements, i.e., from $L = 10$ to 60, when $P_{\max} = 30$ dBm and $\rho = 0.95$. For the secure beamforming approaches assisted by the IRS, their achievable secrecy rates and QoS satisfaction levels significantly increase with the number of the IRS elements. The improvement results from the fact that more IRS elements, more signal paths and signal power can be reflected by the IRS to improve the received SINR at the legitimate ships but to decrease the received SINR at the eavesdropper ship. In addition, the performance of the approach without IRS remains constant under the different numbers of the IRS elements.

From Fig. 2.6a, it is found that the secrecy rate of the proposed learning approach is higher than those of the Baseline 1 and DQN approaches; especially their performance gap also obviously increases with L. This is because with more reflecting elements at the IRS, the proposed deep PDS-PER learning-based secure communication approach becomes more flexible for optimal phase shift (reflecting beamforming) design and hence achieves higher gains. In addition, from Fig. 2.6b compared with the DQN approaches, as the reflecting elements at the IRS increase, we observe that the proposed learning approach is the first one that attains 100%

Fig. 2.6 Performance comparisons versus the number of IRS elements. (**a**) Average secrecy rate. (**b**) QoS satisfaction

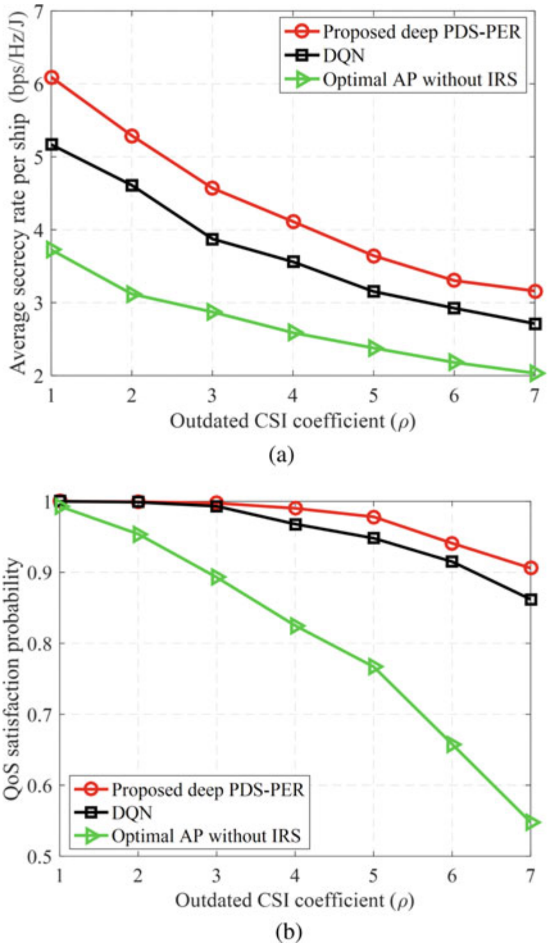


QoS satisfaction level. These superior achievements are based on the particular design of the QoS-aware reward function shown in (2.14) for secure communication.

In Fig. 2.7, we further analyze how the system secrecy rate and QoS satisfaction level performances are affected by the outdated CSI coefficient ρ in the system, i.e., from $\rho = 0.5$ to 1, when $P_{\max} = 30$ dBm and $L = 40$. Note that as ρ decreases, the CSI becomes more outdated as shown in (4) and (6), and $\rho = 1$ means non-outdated CSI. It can be observed from all beamforming approaches, when CSI becomes more outdated (as ρ decreases), that the average secrecy rate and QoS satisfaction level decrease. The reason is that a higher value of ρ indicates more accurate CSI, which will enable all the approaches to optimize secure beamforming policy to achieve higher average secrecy rate and QoS satisfaction level in the system.

It can be observed that reducing ρ has more effects on the performance of the other three approaches, while our proposed learning approach still maintains the

Fig. 2.7 Performance comparisons versus outdated CSI coefficient ρ. (**a**) Average secrecy rate. (**b**) QoS satisfaction



performance at a favorable level, indicating that the other three approaches are more sensitive to the uncertainty of CSI and the robustness of the proposed learning approach. For instance, the proposed learning approach achieves a secrecy rate and QoS satisfaction level improvements of about 17% and 9%, compared with the Baseline 1 approach when $\rho = 0.7$. Moreover, in comparison, the proposed learning approach achieves the best performance among all approaches against channel uncertainty. The reason is that the proposed learning approach considers the time-varying channels and takes advantage of PDS learning to effectively learn the dynamic environment.

2.6 Conclusion

In this work, we have investigated the joint AP's beamforming and IRS's reflecting beamforming optimization problem under the time-varying channel conditions in maritime wireless communications. As the system is highly dynamic and complex, we have exploited the recent advances of machine learning and formulated the secure beamforming optimization problem as an RL problem. A deep PDS-PER learning-based secure beamforming approach has been proposed to jointly optimize both the AP's beamforming and the IRS's reflecting beamforming in the dynamic IRS-aided secure communication system, where PDS and PER schemes have been utilized to improve the learning convergence rate and efficiency. Simulation results have verified that the proposed learning approach outperforms other existing approaches in terms of enhancing the system secrecy rate and the QoS satisfaction probability for maritime wireless communications.

References

1. Y. Liu, C. X. Wang, H. Chang, Y. He, J. Bian, A novel non-stationary 6G UAV channel model for maritime communications. IEEE J. Sel. Areas Commun. **39**(10), 2992–3005 (2021)
2. Y. Shi, L. Zheng, W. Lin, X. Ma, Spatial-modulated physical-layer network coding based on block Markov superposition transmission for maritime relay communications. China Commun. **17**(3), 26–35 (2020)
3. R. Duan, J. Wang, H. Zhang, Y. Ren, L. Hanzo, Joint multicast beamforming and relay design for maritime communication systems. IEEE Trans. Green Commun. Netw. **4**(1), 139–151 (2020)
4. T. Yang, H. Liang, N. Cheng, R. Deng and X. Shen, Efficient scheduling for video transmissions in maritime wireless communication networks. IEEE Trans. Veh. Technol. **64**(9), 4215–4229 (2015)
5. W. Wang, K. C. Teh and K. H. Li, Artificial noise aided physical layer security in multi-antenna small-cell networks. IEEE Trans. Inf. Forensics Secur. **12**(6), 1470–1482 (2017)
6. H. Wang, T. Zheng, X. Xia, Secure MISO wiretap channels with multiantenna passive eavesdropper ship: Artificial noise vs. artificial fast fading. IEEE Trans. Wireless Commun. **14**(1), 94–106 (2015)
7. R. Nakai, S. Sugiura, Physical layer security in buffer-state-based max-ratio relay selection exploiting broadcasting with cooperative beamforming and jamming. IEEE Trans. Inf. Forensics Secur. **14**(2), 431–444 (2019)
8. K. Liu, P. Li, C. Liu, L. Xiao and L. Jia, UAV-aided anti-jamming maritime communications: A deep reinforcement learning approach, in *Proceedings of the 13th International Conference on Wireless Communications and and Signal Processing (WCSP)*, Changsha, China, Dec. 2021
9. Q. Wu, R. Zhang, Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network. IEEE Commun. Mag. **58**(1), 106–112 (2020)
10. J. Zhao, A survey of intelligent reflecting surfaces (IRSs): Towards 6G wireless communication networks, 2019. [Online]. Available: https://arxiv.org/abs/1907.04789
11. H. Han. et al., Intelligent reflecting surface aided network: Power control for physical-layer broadcasting, in *Proceedings of the IEEE International Conference on Communications (ICC)*, Dublin, Ireland, Jul. 2020

12. C. Huang, A. Zappone, G.C. Alexandropoulos, M. Debbah, C. Yuen, Reconfigurable intelligent surfaces for energy efficiency in wireless communication. IEEE Trans. Wireless Commun. **18**(8), 4157–4170 (2019)
13. Q. Wu, R. Zhang, Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming. IEEE Trans. Wireless Commun. **18**(11), 5394–5409 (2019)
14. M. Cui, G. Zhang, R. Zhang, Secure wireless communication via intelligent reflecting surface. IEEE Wireless Commun. Lett. **8**(5), 1410–1414 (2019)
15. H. Shen, W. Xu, S. Gong, Z. He, C. Zhao, Secrecy rate maximization for intelligent reflecting surface assisted multi-antenna communications. IEEE Commun. Lett. **23**(9), 1488–1492 (2019)
16. X. Yu, D. Xu, R. Schober, Enabling secure wireless communications via intelligent reflecting surfaces, in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, HI, Dec. 2019
17. Q. Wu, R. Zhang, Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts. IEEE Trans. Commun. **68**(3), 1838–1851 (2020)
18. Z. Chu, W. Hao, P. Xiao, J. Shi, Intelligent reflecting surface aided multi-antenna secure transmission. IEEE Wireless Commun. Lett. **9**(1), 108–112 (2020)
19. B. Feng, Y. Wu, M. Zheng, Secure transmission strategy for intelligent reflecting surface enhanced wireless system, in *Proceedings of the 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, Xi'an, China, Dec. 2019
20. J. Chen, Y. Liang, Y. Pei, H. Guo, Intelligent reflecting surface: A programmable wireless environment for physical layer security. IEEE Access **7**, 82599–82612 (2019)
21. X. Yu, D. Xu, Y. Sun, D.W.K. Ng, R. Schober, Robust and secure wireless communications via intelligent reflecting surfaces. IEEE J. Sel. Areas Commun. **38**(11), 2637–2652 (2020)
22. X. Guan, Q. Wu, R. Zhang, Intelligent reflecting surface assisted secrecy communication: Is artificial noise helpful or not? IEEE Wireless Commun. Lett. **9**(6), 778–782 (2020)
23. L. Dong, H. Wang, Secure MIMO transmission via intelligent reflecting surface. IEEE Wireless Commun. Lett. **9**(6), 787–790 (2020)
24. W. Jiang, Y. Zhang, J. Wu, W. Feng, Y. Jin, Intelligent reflecting surface assisted secure wireless communications with multiple-transmit and multiple-receive antennas. IEEE Access **8**, 86659–86673 (2020)
25. D. Xu, X. Yu, Y. Sun, D.W.K. Ng, R. Schober, Resource allocation for secure IRS-assisted multiuser MISO systems, in *Proceedings of the IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, HI, Dec. 2019
26. C. Huang, G.C. Alexandropoulos, C. Yuen, et al., Indoor signal focusing with deep learning designed reconfigurable intelligent surfaces, in *Proceedings of the IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Cannes, France, 2019
27. A. Taha, M. Alrabeiah, A. Alkhateeb, Enabling large intelligent surfaces with compressive sensing and deep learning. IEEE Access **9**, 44,304–44,321 (2021)
28. K. Feng, Q. Wang, X. Li, C. Wen, Deep reinforcement learning based intelligent reflecting surface optimization for MISO communication systems. IEEE Wireless Commun. Lett. **9**(5), 745–749 (2020)
29. C. Huang, R. Mo, C. Yuen, Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning. IEEE J. Sel. Areas Commun. **38**(8), 1839–1850 (2020)
30. C. Li, W. Zhou, K. Yu, L. Fan, J. Xia, Enhanced secure transmission against intelligent attacks. IEEE Access **7**, 53596–53602 (2019)
31. L. Xiao, G. Sheng, S. Liu, H. Dai, M. Peng, J. Song, Deep reinforcement learning-enabled secure visible light communication against eavesdropping. IEEE Trans. Commun. **67**(10), 6994–7005 (2019)
32. M. Wiering, M. Otterlo, *Reinforcement Learning: Stateof-the-Art* (Springer Berlin, Heidelberg, 2014)

33. H.L. Yang, A. Alphones, C. Chen, W.D. Zhong, X.Z. Xie, Learning-based energy-efficient resource management by heterogeneous RF/VLC for ultra-reliable low-latency industrial IoT networks. IEEE Trans. Ind. Inf. **16**(8), 5565–5576 (2020)
34. X. He, R. Jin, H. Dai, Deep PDS-learning for privacy-aware offloading in MEC-enabled IoT. IEEE Internet Things J. **6**(3), 4547–4555 (2019)
35. N. Mastronarde, M. van der Schaar, Joint physical-layer and systemlevel power management for delay-sensitive wireless communications. IEEE Trans. Mobile Comput. **12**(4), 694–709 (2013)
36. T. Schaul, J. Quan, I. Antonoglou, D. Silver, Prioritized experience replay, in *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, San Juan, Colorado, May. 2016
37. H. Gacanin, M. Di Renzo, Wireless 2.0: Towards an intelligent radio environment empowered by reconfigurable meta-surfaces and artificial intelligence. IEEE Veh. Technol. Mag. **15**(4), 74–82 (2020)
38. C.W. Huang, et al., Holographic MIMO surfaces for 6G wireless networks: Opportunities, challenges, and trends. IEEE Wireless Commun. **27**(5), 118–125 (2020)
39. F.B. Mismar, B.L. Evans, A. Alkhateeb, Deep reinforcement learning for 5G networks: Joint beamforming, power control, and interference coordination. IEEE Trans. Commun. **68**(3), 1581–1592 (2020)

Chapter 3
Learning-Based Privacy-Aware Maritime IoT Communications

Mobile edge computing helps maritime IoT devices with energy harvesting to provide satisfactory experiences for computation-intensive applications in maritime communication systems, such as real-time cargo status notification, emergency rescue in maritime affairs, and accurate early warning. In this chapter, we present an RL-based privacy-aware offloading scheme to help maritime IoT devices protect both the user location privacy and the usage pattern privacy. More specifically, this scheme enables a maritime IoT device to choose the offloading rate that improves the computation performance, protects user privacy, and saves the energy of the IoT devices without being aware of the privacy leakage, energy consumption, and edge computation model. This scheme uses transfer learning to reduce the random exploration at the initial learning process and applies a Dyna architecture that provides simulated offloading experiences to accelerate the learning process. Furthermore, a post-decision state learning method uses the known channel state model to improve the offloading performance. The performance bound of this scheme is provided regarding the privacy level, the energy consumption, and the computation latency for three typical ship offloading scenarios.

3.1 Introduction

The main techniques used in this chapter are introduced first, including MEC and energy harvesting (EH). Then, the privacy issue in MEC IoT is highlighted.

L. Xiao et al., *Reinforcement Learning for Maritime Communications*, Wireless Networks, https://doi.org/10.1007/978-3-031-32138-2_3

3.1.1 Mobile Edge Computing

With the rapid development of IoT, the number of mobile terminal devices and the amount of sensing data are increasing exponentially, which puts forward new requirements for the IoT devices' storage and computing capabilities. MEC technology helps IoT devices support computational-intensive and latency-sensitive applications with reduced energy consumption and computation latency, regarded as one of the key technologies of 5G.

As shown in Fig. 3.1, a MEC system mainly includes mobile devices (such as ships, end users, clients, service subscribers, etc.) and edge devices. Edge devices are typically small data centers like small base stations, access points, or laptops. MEC is widely used in smart device applications, maritime IoT, medical and health monitoring systems, Internet of vehicles, and monitoring networks. IoT devices offload local computation tasks to the MEC network to solve the shortage of local resource storage and computation performance of IoT devices, which has a significant effect on reducing computation latency, energy consumption, and saving network bandwidth, to meet the requirements of scenarios with lower latency and higher bandwidth [1]. For instance, the binary offloading as proposed in [2] chooses the data transmission rate under a stochastic wireless channel with a single edge to reduce the computation overhead for resource-constrained mobile devices. The partial offloading scheme as proposed in [3] uses the time-division and the orthogonal frequency-division multiple access to reduce the energy consumption under a latency constraint in a multiuser MEC network. The mobile offloading

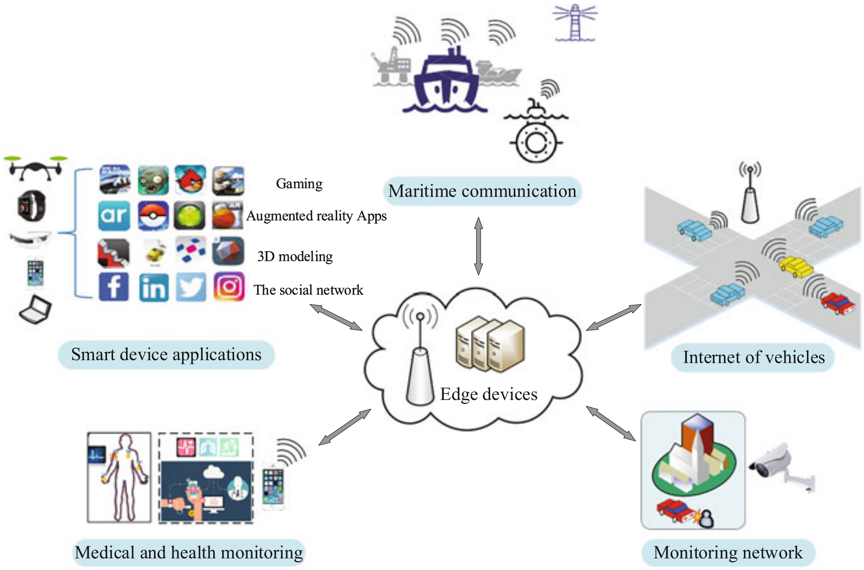


Fig. 3.1 System architecture of mobile edge computing network

scheme, as proposed in [4], uses the Lyapunov optimization to reduce the execution latency and the task failure rate for the case with a single known MEC server, assuming that both the transmission delay model and the local execution model are known.

The application of MEC technology in implementing different IoT systems is summarized in a comprehensive survey [5] and analyzes how MEC can improve the performance of IoT networks. In popular scenarios such as the Internet of vehicles and the industrial IoT, the intelligent offloading framework is used to offload the computation tasks from a single IoT device to multiple edge server devices, and the task allocation accompanied by CPU frequency is jointly optimized to minimize the execution latency and energy consumption [6]. For the maritime IoT scenario, a novel two-stage offloading optimization for energy-latency tradeoff with MEC in maritime IoT is investigated in [7], providing an efficient guideline for optimizing the maritime communication networks. In addition, the risk of data privacy leakage of a maritime mobile terminal (MMT) during the offloading tasks is analyzed in [8], and an attribute sensitivity-based differential privacy (AS-DP) algorithm to balance the security and availability of data is proposed.

3.1.2 Energy Harvesting

EH is a promising technique to prolong the battery lifetime and provide a satisfactory experience for IoT devices [9]. In the traditional IoT system, using the power grid will inevitably lead to a large number of carbon emissions, which does not meet the needs of energy conservation and emission reduction. Thanks to the rapid development of EH technology, renewable energy (such as solar energy [10], water wave [11], wind energy [12], and ambient radio-frequency signals [13]) has become a promising source of power supply for various information systems in recent years. Additionally, using renewable energy for power also reduces manual interventions such as battery replacement/charging, especially for hazardous or hard-to-reach applications.

EH technology has facilitated the development of MEC systems powered by renewable energy, as shown in Fig. 3.2, which includes renewable energy-powered edge devices and mobile IoT devices. An optimization scheme for EH and task calculation based on a differential evolution algorithm is proposed in [14], which not only has higher optimization efficiency and low energy consumption but also can effectively alleviate the energy shortage of micro devices and extend their service life. A renewable-powered MEC system combines the value iteration with the RL technique to improve the edge computing performance of the mobile device for delay-sensitive applications with intermittent and unpredictable renewable energy [15]. The wireless-powered multiuser MEC system as proposed in [16] jointly improves the AP beamforming and the user time allocation to save the AP energy consumption subject to the users' latency constraints.

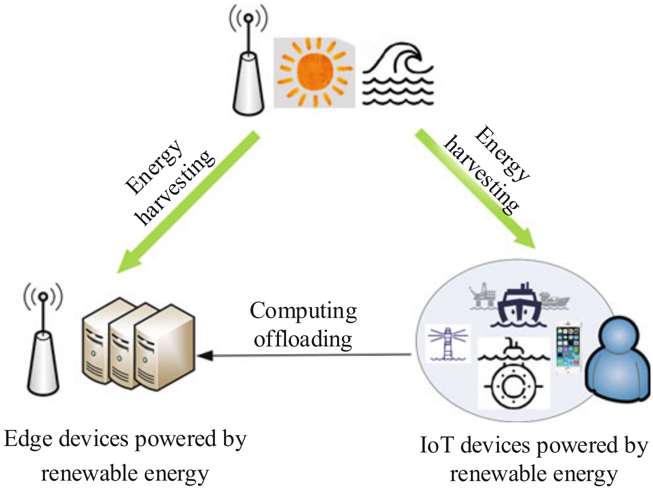

Energy harvesting
Energy harvesting
Computing offloading
Edge devices powered by renewable energy
IoT devices powered by renewable energy

Fig. 3.2 Mobile edge computing network based on energy harvesting technology

A self-sustaining broadband long-range maritime communication system is proposed in [17], where the EH unit generates electrical energy from ocean waves to support the operation of the wireless communication unit. Moreover, tidal energy is captured and powered for the underwater IoT networks [11]. In addition, the ocean thermal energy associated with vertical temperature differentials between the warm surface and cold deep water can potentially be a sustainable power for autonomous underwater vehicles and sensors indefinitely [18].

The fundamental problem that needs to be solved for a MEC system powered by renewable energy is green energy awareness resource allocation and computation offloading. In the case of renewable energy power supply, the design principle of MEC systems is no longer to minimize energy consumption to satisfy user experience requirements, but to optimize system performance under the given energy constraints. In addition, for mobile edge computing systems powered by renewable energy, the production capacity of renewable energy plays a crucial role in system decision-making. Due to the intermittent and unpredictable nature of renewable energy, the capacity situation changes with the external environment (such as weather), which challenges the formulation of computation offloading strategies for MEC systems (such as unreliable computation offloading or the risk of task computation failure).

3.1.3 Privacy in MEC IoTs

The popularization and application of MEC technology in the IoT environment bring new challenges of privacy leakage. To meet the computation requirements

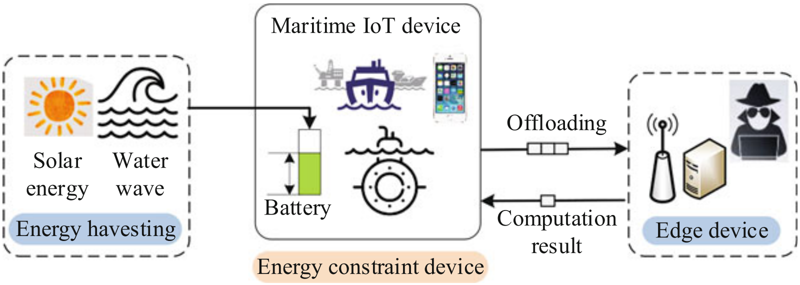


Fig. 3.3 The application of MEC in maritime scenario and its privacy threat

of resource-constrained IoT devices for task-intensive and latency-sensitive tasks (such as augmented reality/virtual reality, AR/VR), MEC has become a promising technology [1, 19]. IoT devices offload computation tasks to edge devices to improve task computation efficiency, reduce computation latency, and extend the IoT devices' lifetime. For example, as shown in Fig. 3.3, the maritime IoT devices offload the locally generated sensing marine data to the surrounding edge devices. The edge device assists in the task computation and feedbacks the computation results to the IoT users on time, providing timely and effective maritime emergency accident diagnosis. However, when users upload local data to the edge device via wireless communication channels, not fully trusted/honest but curious edge devices or eavesdropping attackers can analyze or reuse users' data for illegal financial gain [20], so user privacy is at risk. Therefore, it is crucial to ensure user privacy and security while using MEC technique [21]. Currently, most of the research on MEC offloading focuses on improving the computation performance of the system while ignoring privacy protection [4, 15].

Several papers point out that privacy protection is critical for edge computing of IoT applications [21–24]. For instance, the IoT computation offloading scheme as proposed in [25, 26] uses steganography and homomorphic encryption to hide the image privacy and save energy and protect privacy, respectively. A privacy-preserving opportunistic computing framework for m-Healthcare emergence as proposed in [27] exploits the attribute-based access control and the privacy-preserving scalar product computation technique to reduce the medical data privacy disclosure.

3.2 Related Work

IoT devices are widely used in maritime communication systems such as ship-to-shore connectivity to provide real-time cargo status notification, emergency rescue in maritime affairs, and accurate early warning [28]. Maritime IoT devices can apply the EH technique to use the energy from the environment, such as solar energy, wind energy, and water wave, to extend the battery life [9, 29]. MEC also saves energy for IoT devices by processing the maritime monitoring data at the edge devices, such as the BSs and other neighbor ships that have more computation and energy resources [30–34]. The space-air-ground-edge (SAGE) maritime communication network architecture was first proposed in [35], which was used to offload computing-intensive applications and services for IoT users in the marine environment. For instance, an edge device can help the user in a ship evaluate the monitored emergency data and make effective operations.

Maritime IoT devices with EH have to resist eavesdroppers that analyze the maritime sensing data via radio channels to reveal the user location and characteristics, such as the usage pattern privacy [22, 23, 36, 37]. More specifically, the user location privacy can be inferred from the offloading data size, e.g., a user is very likely to stay in the outage locations or far away from the edge device if the IoT device locally computes all the maritime sensing data under severe radio channel condition connecting to the edge device. An attacker can estimate the size of the maritime sensing data newly generated and thus evaluate the usage pattern if the IoT device offloads all the maritime sensing data to the edge device under a good radio channel state. Therefore, the IoT device must protect both the user location and usage pattern privacy in mobile offloading.

Current steganography and homomorphic encryption techniques are not always applicable for maritime IoT devices with limited computation resources during the edge computing [25, 26], and most existing mobile offloading schemes such as [4, 15, 16] ignore user privacy. The seminar work on the privacy-aware mobile computing as presented in [20] allows mobile devices to choose the offloading policy and formulates a constrained Markov decision process (CMDP) to ensure the pre-specified privacy level for a simplified offloading scenario with reduced computation latency and energy consumption. This scheme suffers from a slow learning speed and the offloading performance degrades in practical maritime IoT devices with energy harvesting.

Reinforcement learning techniques have been applied for offloading in MEC. For instance, a Q-learning-based traffic offloading scheme as presented in [38] makes a trade-off between energy consumption and the quality of service for mobile devices in heterogeneous cellular networks. An online learning-based resource management algorithm in [15] uses the post-decision state to choose the on-the-fly workload offloading rate to both the centralized cloud and the edge server to reduce both the service delay and the operational cost. The computation offloading strategy proposed in [39] uses Q-learning to help IoT devices choose the offloading rate and reduce the attack rate of smart attackers without being aware of the channel model. DRL can be considered as an advanced RL technique implemented with DNNs, by exploiting the function approximation property, faced with offloading policy selection for high-dimensional and continuous computing of IoT devices of DNNs. A novel offloading strategy based on the deep Q network (DQN) was designed in [40] to study a multiuser MEC network, where tasks from users can be partially offloaded to multiple computational access points (CAPs). Moreover, the hybrid DQN was proposed to solve the mixed-integer non-convex problem in [41], whose performance was demonstrated better than the pure-DQN approach.

In this chapter, we present a privacy-aware offloading scheme to improve the privacy level, reduce the computation latency, and save the energy consumption of maritime IoT devices. In this scheme, the offloading rate and the local processing rate of an IoT device are chosen based on the current radio channel state, the size and priority of the maritime sensing data or computation tasks, the estimated energy harvesting state, and the battery level. For instance, more maritime sensing data are offloaded to the edge device under a good radio channel state. Otherwise, the IoT device locally processes the maritime sensing data in rare cases with narrow radio bandwidth to save computation latency. Therefore, the presented offloading scheme optimizes the offloading rate according to the radio channel state to improve the computation performance of the IoT device. This scheme analyzes the difference between the amount of maritime sensing data and the size of the offloading data under different channel power gains to avoid privacy leakage.

The optimal offloading policy depends on the accurate knowledge of the privacy leakage, the energy consumption, and the edge computation model in each time slot, which is challenging to determine especially in dynamic maritime IoT communication systems. As the future state observed by a maritime IoT device is independent of the previous states for a given current state and offloading policy in a repeated offloading process, we have an MDP, and thus a maritime IoT device can apply RL techniques, such as Q-learning to achieve the optimal offloading policy via trial and error without being aware of the underlying models [42]. We present an RL-based privacy-aware offloading algorithm for a maritime IoT device to choose the offloading and local computing policy. This offloading algorithm uses the model learning method that exploits the offloading experiences to build a Dyna architecture and generate simulated experiences accordingly to update the value function of the reinforcement learning technique. A PDS method as investigated in [43] is also applied to use the known radio channel model to accelerate the learning process. A transfer learning method as developed in [44] is used to exploit the offloading experiences in similar scenarios for the initialization of the learning parameters and thus save the initial exploration in the offloading process.

We prove that the presented scheme achieves the optimal offloading policy after long enough time slots in the dynamic game. The offloading performance bound is provided in terms of the privacy level, the total computation latency, and the energy consumption of the maritime IoT device with EH. This offloading algorithm can improve the privacy level of the ship, which depends on the amount of computation tasks. Both the computation latency and the energy consumption of the IoT device linearly increase with the size of the maritime sensing data. More specifically, the major contribution of this chapter is summarized as follows:

1. A privacy-aware offloading scheme for an EH-powered maritime IoT device is investigated to select the offloading rate and the local computation rate to process the maritime sensing data. This scheme considers the current radio channel state, the size and the priority of the new maritime sensing data, the estimated energy

harvesting state, the battery level, and the task computation history to decrease the computation latency, save the energy consumption of the IoT device, and improve the privacy level.

2. We present an RL-based offloading algorithm for an IoT device to achieve the optimal offloading policy via trial and error without being aware of the privacy leakage, energy consumption, and edge computation model. This algorithm uses the transfer learning technique, the PDS method, and the Dyna architecture to accelerate the learning process of an IoT device.
3. We provide the performance bound of the RL-based offloading scheme in terms of the privacy level, energy consumption, and computation latency and prove its convergence to the optimal performance in the dynamic offloading process.

The remainder of this chapter is organized as follows. Chapter 3.3 presents the system model. The privacy leakage threats and protection schemes are illustrated in Chap. 3.4. We present an RL-based privacy-aware offloading scheme for maritime IoT devices and analyze its performance in Chap. 3.5. We provide simulation results in Chap. 3.6 and conclude this work in Chap. 3.7.

3.3 System Model

We consider a maritime IoT device that uses multiple sensors to measure and evaluate the maritime data, such as the movement of ships near the shore and the ship performance data to provide real-time cargo status notification, emergency rescue in maritime affairs, and accurate early warning. Powered with both the battery and the energy harvesting module, the IoT device can locally process some computation tasks, offload some tasks to the edge device, and save the others to process in the next time slot, as shown in Fig. 3.4.

The IoT device at time slot k is assumed to generate new maritime sensing data of size $C_1^{(k)}$ and has to process the previous sensing data stored in the buffer of size $C_0^{(k)}$. The time index k in the superscript is omitted if no confusion occurs. By applying the computation partition scheme proposed in [45], the maritime IoT device divides the sensing data of size $C_1^{(k)} + C_0^{(k)}$ into N equivalent computation tasks for simplicity. The priority of such sensing data denoted by $\chi^{(k)}$ can be estimated according to the data analysis algorithm [46].

The IoT device offloads $x_1^{(k)}$ the computation tasks to the edge device over the radio channel with the radio channel power gain $h^{(k)}$, locally processes $x_0^{(k)}$ of the sensing data with the local CPU at the computation speed of f bits per second, and stores the rest of the tasks in the buffer to process in the future, with $\{x_0^{(k)}, x_1^{(k)}\} \in \{l_0/N, l_1/N\}_{0\le l_0,l_1\le N}$. The radio channel power gain $h^{(k)}$ is formulated as a Markov chain model with

$$\Pr\left(h^{(k+1)} = m|h^{(k)} = n\right) = h_{mn}, \forall\, m, n \in \mathbf{H}, \tag{3.1}$$

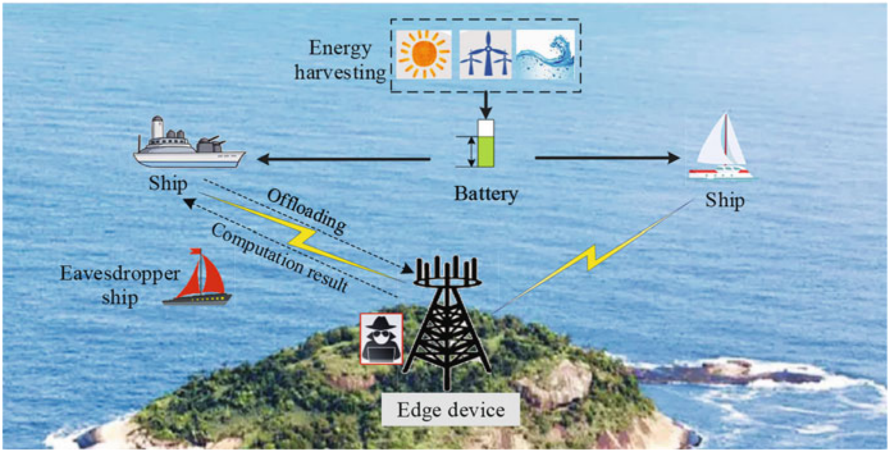


Fig. 3.4 Illustration of the privacy-aware offloading of the EH-powered maritime IoT device (e.g., ship)

where **H** is the radio channel state set. The IoT device consumes ς energy to process one bit sensing data and uses P energy to send one bit sensing data to the edge device.

The edge device sends the computation results to the IoT device. An attacker (e.g., edge device or eavesdropper ship) might be curious about the user privacy, such as the user location and the usage pattern of the IoT device. An edge device can infer the location privacy and the usage pattern of the IoT device based on the offloading history under different channel states that depend on the distance between the user and the edge node [20]. The privacy level is associated with the size of sensing data and the offloading rate.

The IoT device applies the privacy metric similar to [20] to evaluate the privacy level denoted by $R^{(k)}$, estimate the queuing cost denoted by $W^{(k)}$, and measure the computation latency denoted by $T^{(k)}$ and the energy consumption denoted by $E^{(k)}$. The computation latency is the maximum of the local processing latency $T_0^{(k)}$ and the processing delay of offloading $T_1^{(k)}$, i.e.,

$$T = \max\left\{T_0^{(k)}, T_1^{(k)}\right\}. \tag{3.2}$$

The energy consumption of the IoT device consists of the local processing energy consumption $E_0^{(k)}$ and the transmission cost $E_1^{(k)}$.

The solar energy harvester and piezoelectric materials enable the IoT device to convert renewable energy (such as solar energy, wind energy, and water wave) to electricity [29, 47, 48]. The maritime IoT device obtains $\rho^{(k)}$ harvested energy at time slot k to support the local data processing and offloading. The battery level at the beginning of time slot k denoted by $b^{(k)}$ is related to the previous battery level, the energy consumption, and the harvested energy given by

Table 3.1 List of notations

Symbol	Description
$C_1^{(k)}$	Amount of the maritime sensing data newly generated at time slot k
$\chi^{(k)}$	Priority of the maritime data
$h^{(k)}$	Channel power gain between the maritime IoT device and the edge device
$\rho^{(k)}$	Amount of the harvested energy
$C_0^{(k)}$	Amount of the maritime data in the buffer
$b^{(k)}$	Battery level
$\mathbf{x}^{(k)} \in \mathbf{A}$	Offloading strategy
P	Energy consumption per bit for offloading the maritime data to the edge device
ς	Energy consumption for the maritime IoT device to process a bit data
f	Computation capability of the maritime IoT device
$R^{(k)}$	Achieved privacy level
$W^{(k)}/\hat{W}^{(k)}$	Actual/measured queuing cost
$E^{(k)}/\hat{E}^{(k)}$	Actual/measured energy consumption
$T^{(k)}/\hat{T}^{(k)}$	Actual/measured computation latency

$$b^{(k)} = b^{(k-1)} - E^{(k-1)} + \rho^{(k-1)}. \tag{3.3}$$

Note that the IoT device drops the computation tasks at time slot k, if its energy is insufficient, i.e., $b^{(k+1)} < 0$ [4]. Important symbols are summarized in Table 3.1.

3.4 Privacy in MEC

In this section, we introduce the privacy issues in MEC, including location privacy and usage pattern privacy. After that, we provide a privacy protection scheme by adjusting the offloading policy.

3.4.1 Privacy Issues in MEC

In the maritime IoT environment, the maritime IoT device uses the MEC technology to perform task computing and sends the computation task to the edge device through a wireless communication channel. In general, in order to reduce the energy consumption and computation latency in the offloading process, when the wireless communication channel status is good, the IoT device will offload all the sensing data to the edge device; when the wireless communication channel status is poor, the IoT device will choose to process all the sensing data locally to reduce computation latency.

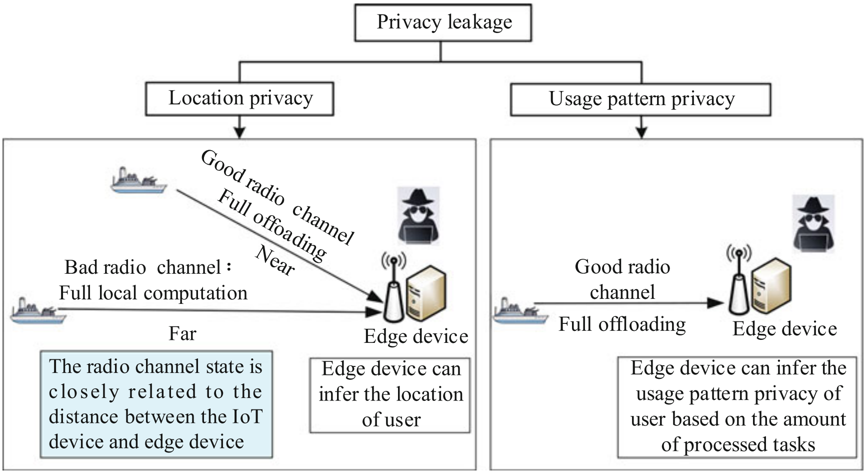

Privacy leakage
Location privacy
Usage pattern privacy
Good radio channel
Full offloading
Near
Bad radio channel :
Full local computation
Far
Edge device
The radio channel state is closely related to the distance between the IoT device and edge device
Edge device can infer the location of user
Good radio channel
Full offloading
Edge device
Edge device can infer the usage pattern privacy of user based on the amount of processed tasks

Fig. 3.5 Illustration of the privacy leakage in MEC

However, this simple computation offloading mechanism exposes the private information of IoT users. As shown in Fig. 3.5, an untrusted edge device or eavesdropping attacker can obtain the sensing data of the maritime IoT device (such as a ship user) and use big data analysis technology and traffic analysis technology to infer the user's usage pattern privacy based on the user's task volume [20]. In addition, as shown in Fig. 3.5, the attacker can infer the channel state between the IoT device and the edge device according to the amount of tasks that assist the IoT device in processing and further locate the IoT device [19]. Therefore, IoT devices' location privacy is at risk. Specifically, in the process of MEC offloading, when the IoT device is closer to the edge device, the channel fading is small, and the communication channel state is better, and then the IoT device chooses to offload all computation tasks to the edge device for processing. When the IoT device is far away from the edge device, the channel fading is serious, and the communication channel state is poor. At this time, the IoT device chooses to process all tasks locally. Therefore, the edge device can infer the state of its communication channel according to the size of the offloading task of the IoT device and use this to infer the location of the IoT device. The above computation offloading policy selection is only considered from the perspective of optimizing computation latency and reducing computation energy consumption while ignoring the location privacy leakage of IoT users.

3.4.2 Location and Usage Pattern Privacy Protection

The privacy protection scheme developed in this chapter optimizes the computation task offloading rate based on the wireless communication channel state and improves the computation performance of maritime IoT devices. This scheme can reduce the attacker's information inference accuracy by adjusting the difference between the amount of sensing data and the amount of offloading data under different communication channel states, protecting location privacy and usage pattern privacy. As shown in Fig. 3.6, when the channel state continues to be in a good state (i.e., $h > \hat{h}$, with $\hat{h}$ be the good channel index), the maritime IoT device will not offload all computation tasks to the edge device, but will choose to process some tasks locally. As shown in Fig. 3.7, when the channel state is relatively poor (i.e., $h < \check{h}$, with $\hat{h}$ be the bad channel index), maritime IoT devices will not process all tasks locally, but choose to offload some tasks to the edge device. Through the above methods, the attacker cannot know the amount of tasks actually sensed by the maritime IoT device and the processing status of the tasks and thus cannot obtain the user's private information through traffic analysis or other inference methods.

However, the above privacy protection scheme design will inevitably increase the computation delay and energy consumption of maritime IoT devices, thereby affecting the overall computation efficiency of the maritime IoT system. Therefore, in order to balance the privacy protection requirements of maritime IoT devices with energy consumption and computation latency, we propose a privacy-aware offloading scheme based on reinforcement learning. This scheme can achieve the optimal offloading policy via trial and error without being aware of the underlying models, which can reduce the computation latency, save energy consumption, and improve the privacy level of the maritime IoT device.

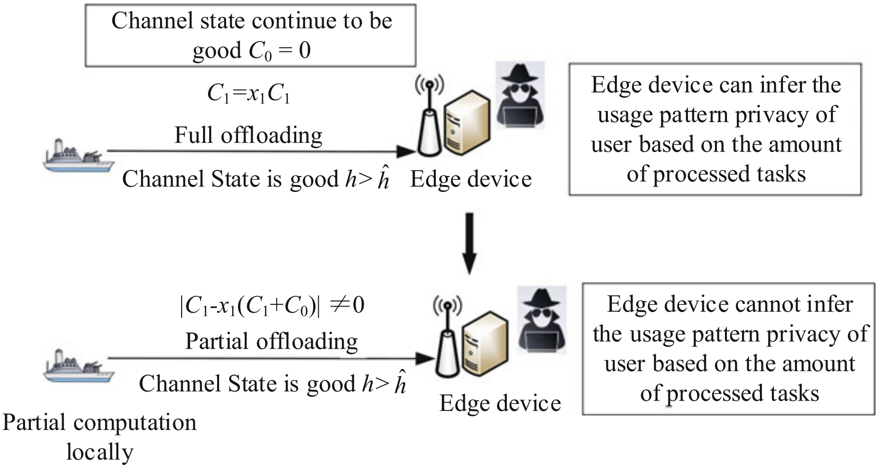

Channel state continue to be good $C_0 = 0$
$C_1 = x_1 C_1$
Full offloading
Channel State is good $h > \hat{h}$
Edge device
Edge device can infer the usage pattern privacy of user based on the amount of processed tasks
$|C_1 - x_1(C_1 + C_0)| \neq 0$
Partial offloading
Channel State is good $h > \hat{h}$
Edge device
Edge device cannot infer the usage pattern privacy of user based on the amount of processed tasks
Partial computation locally

Fig. 3.6 Illustration of the usage pattern privacy protection in MEC

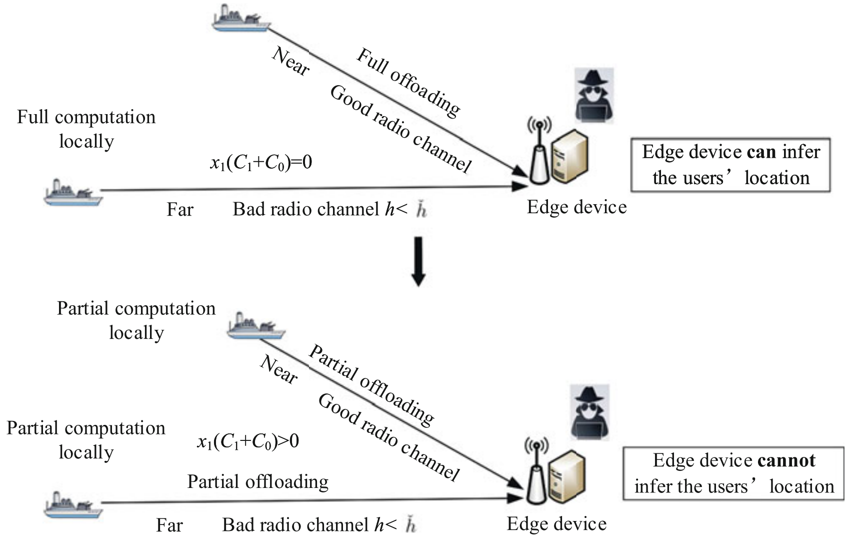

Near
Full offloading
Good radio channel
Full computation locally
$x_1(C_1+C_0)=0$
Edge device can infer the users' location
Far
Bad radio channel $h<\tilde{h}$
Edge device
Partial computation locally
Near
Partial offloading
Good radio channel
Partial computation locally
$x_1(C_1+C_0)>0$
Partial offloading
Edge device cannot infer the users' location
Far
Bad radio channel $h<\tilde{h}$
Edge device

Fig. 3.7 Illustration of the location privacy protection in MEC

3.5 Learning-Based Privacy-Aware Offloading with Energy Harvesting

We present an RL-based privacy-aware offloading scheme, as shown in Fig. 3.8, for a maritime IoT device to choose both the offloading rate and the local processing rate. More specifically, the offloading policy is chosen based on the expected discounted long-term utility or Q-function denoted by Q for the current state. The offloading policy is chosen based on the current state $\mathbf{s}^{(k)}$ that consists of the size and priority of the new maritime sensing data, the current radio channel state, the estimated renewable energy generated in the time slot, the current battery level of the IoT device, and the computation history. This scheme applies the known radio channel model and generates simulated experiences to reduce the time required to learn the optimal policy.

3.5.1 Privacy-Aware Offloading

As shown in Algorithm 2, upon measuring the maritime data of size $C_1^{(k)}$ at time slot k, a maritime IoT device briefly evaluates the priority of the maritime data denoted by $\chi^{(k)}$ and estimates the channel power gain to the edge device $h^{(k)}$. According to the historical record and the offloading experiences, the IoT device estimates the amount of the harvested energy $\hat{\rho}^{(k)}$ and observes the current battery level of the

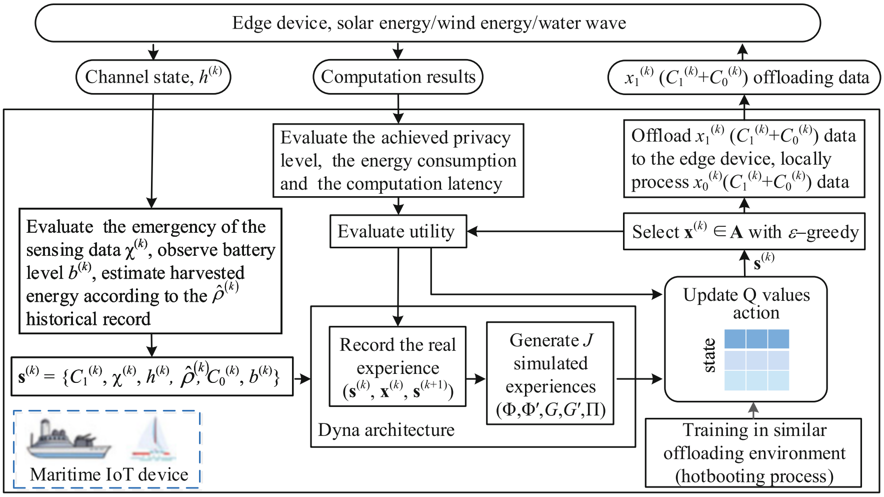

Edge device, solar energy/wind energy/water wave
Channel state, $h^{(k)}$
Computation results
$x_1^{(k)}$ $(C_1^{(k)}+C_0^{(k)})$ offloading data
Evaluate the achieved privacy level, the energy consumption and the computation latency
Offload $x_1^{(k)}$ $(C_1^{(k)}+C_0^{(k)})$ data to the edge device, locally process $x_0^{(k)}(C_1^{(k)}+C_0^{(k)})$ data
Evaluate the emergency of the sensing data $\chi^{(k)}$, observe battery level $b^{(k)}$, estimate harvested energy according to the $\hat{\rho}^{(k)}$ historical record
Evaluate utility
Select $\mathbf{x}^{(k)} \in \mathbf{A}$ with ε–greedy
$\mathbf{s}^{(k)}$
Update Q values
action
state
$\mathbf{s}^{(k)} = \{C_1^{(k)}, \chi^{(k)}, h^{(k)}, \hat{\rho}^{(k)}, C_0^{(k)}, b^{(k)}\}$
Record the real experience $(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, \mathbf{s}^{(k+1)})$
Generate J simulated experiences (Φ,Φ',G,G',Π)
Dyna architecture
Training in similar offloading environment (hotbooting process)
Maritime IoT device

Fig. 3.8 Illustration of the RL-based privacy-aware offloading for maritime IoT devices

IoT device $b^{(k)}$. The state is chosen as $\mathbf{s}^{(k)} = \{C_1^{(k)}, \chi^{(k)}, h^{(k)}, \hat{\rho}^{(k)}, C_0^{(k)}, b^{(k)}\}$. Let Λ be the state space.

The new and buffered maritime data with a size of $C_0^{(k)} + C_1^{(k)}$ are divided into N equivalent computation tasks based on the computation partition method [45]. The offloading policy $\mathbf{x}^{(k)} = [x_0^{(k)}, x_1^{(k)}] \in \mathbf{A}$ is chosen according to the ϵ-greedy policy to make a trade-off between exploration and exploitation [42]. More specifically, the offloading policy that maximizes $Q(\mathbf{s}^{(k)}, \mathbf{x})$ is chosen with $1 - \epsilon$, and other feasible offloading policies are randomly selected with a small probability. The maritime IoT device offloads $x_1^{(k)}(C_1^{(k)} + C_0^{(k)})$ maritime data to the edge device, processes $x_0^{(k)}(C_1^{(k)} + C_0^{(k)})$ of the data locally, and stores the rest in the buffer to be processed in the future.

After receiving the computation report from the edge device and finishing the local processing, the IoT device analyzes the difference between the size of the sensed data and the size of the offloading data, as well as the current channel states, to evaluate the achieved privacy level $R^{(k)}$.

More specifically, the IoT device tends to offload all the data to the edge device under a good radio channel state compared with the good channel index $\hat{h}$. On the other hand, the IoT device processes all the data locally under a bad channel state compared with the bad channel index $\check{h}$. Let ω denote the importance of location privacy over usage pattern privacy. The indicator function denoted by $\mathbb{I}(\cdot)$ equals 1 if the statement is true, and 0 otherwise. The achieved privacy level consists of both the achieved usage pattern privacy and the location privacy. The former is modeled with $|C_1^{(k)} - x_1^{(k)}(C_1^{(k)} + C_0^{(k)})|\mathbb{I}(h^{(k)} \geq \hat{h})$, and the latter is given by $\omega\mathbb{I}(x_1^{(k)}(C_1^{(k)} + C_0^{(k)}) > 0)\mathbb{I}(h^{(k)} \leq \check{h})$. The IoT device achieves the privacy level, ξ if $\check{h} < h^{(k)} < \hat{h}$. Similar to [20], the privacy level $R^{(k)}$ is estimated by

Algorithm 2 RL-based privacy-aware IoT offloading algorithm

```
Initialize $\alpha$, $\gamma$ and $\delta$
Hotbooting process
Set $\mathbf{Q} = \overline{\mathbf{Q}}$, $\Phi = 0$, $\Phi' = 0$, $G' = 0$, $G = 0$, $\Pi = 0$
for $k = 1, 2, 3, \ldots$ do
    Observe $C_1^{(k)}$, $C_0^{(k)}$ and $b^{(k)}$
    Evaluate $\chi^{(k)}$
    Estimate $h^{(k)}$ and $\hat{\rho}^{(k)}$
    $\mathbf{s}^{(k)} = \{C_1^{(k)}, \chi^{(k)}, h^{(k)}, \hat{\rho}^{(k)}, C_0^{(k)}, b^{(k)}\}$
    Divide the maritime data with size of $C_0^{(k)} + C_1^{(k)}$ into $N$ equivalent computation tasks
    Choose $\mathbf{x}^{(k)} = [x_0^{(k)}, x_1^{(k)}] \in \mathbf{A}$ with $\varepsilon$-greedy policy
    Offload $x_1^{(k)}(C_1^{(k)} + C_0^{(k)})$ maritime data to the edge device, process $x_0^{(k)}(C_1^{(k)} + C_0^{(k)})$ of
    the data locally, and store the rest in the buffer
    Evaluate the achieved privacy $\hat{R}^{(k)}$, the total energy consumption $\hat{E}^{(k)}$, and the computation
    latency $\hat{T}^{(k)}$
    Evaluate $u^{(k)}$ via (3.5)
    Estimate $\tilde{\mathbf{s}}^{(k)}$ via (3.6)
    Evaluate $b^{(k+1)}$ via (3.3)
    Update $Q(\tilde{\mathbf{s}}^{(k)}, \mathbf{x}^{(k)})$ via (3.7)
    Update $Q(\mathbf{s}^{(k)}, \mathbf{x}^{(k)})$ via (3.8)
    Formulate the real experience $(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, u^{(k)}, \mathbf{s}^{(k+1)})$
    Update $\Phi' \left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, \mathbf{s}^{(k+1)}\right)$ via (3.9)
    Update $\Phi \left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}\right)$ via (3.10)
    Update the state transition probability function $\Pi \left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, \mathbf{s}^{(k+1)}\right)$ via (3.11)
    Calculate the reward record $G' \left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, \Phi \left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}\right)\right)$ via (3.12)
    Update the reward function $G \left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}\right)$ via (3.13)
    for $j = 1$ to $J$ do
        Randomly select $\left(\bar{\mathbf{s}}^{(j)}, \bar{\mathbf{x}}^{(j)}\right)$
        Select $\bar{\mathbf{s}}^{(j+1)}$ based on $\Pi \left(\mathbf{s}^{(j)}, \mathbf{x}^{(j)}, \mathbf{s}^{(j+1)}\right)$
        Calculate $\hat{u}^{(j)} = G \left(\bar{\mathbf{s}}^{(j)}, \bar{\mathbf{x}}^{(j)}\right)$ via (3.14)
        Update $Q \left(\bar{\mathbf{s}}^{(j)}, \bar{\mathbf{x}}^{(j)}\right)$ via (3.7)
    end for
end for
```

$$R^{(k)} = \left|C_1^{(k)} - x_1^{(k)}\left(C_1^{(k)} + C_0^{(k)}\right)\right| \mathbb{I}\left(h^{(k)} \geq \hat{h}\right) + \omega \mathbb{I}\left(x_1^{(k)}\left(C_1^{(k)} + C_0^{(k)}\right) > 0\right) \mathbb{I}\left(h^{(k)} \leq \check{h}\right) + \xi \mathbb{I}\left(\check{h} < h^{(k)} < \hat{h}\right). \tag{3.4}$$

An IoT device deliberately reduces the offloading rate under a good channel state and increases the offloading rate under low channel power gains to protect privacy. The usage pattern privacy as indicated in the first term of (3.4) represents the difference between the actual sensing data size and the offloading data size under high radio channel power gains. The location privacy as indicated in the third term

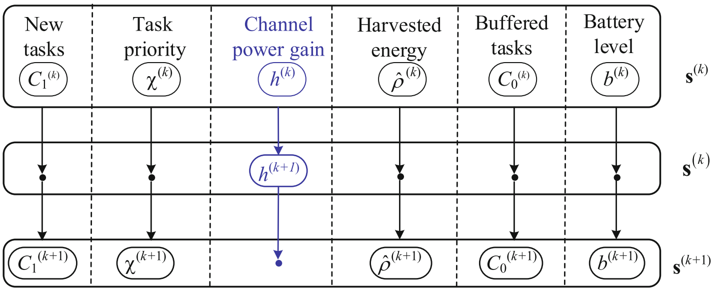


Fig. 3.9 State transition of the maritime IoT device based on PDS learning in the dynamic maritime communication system

of (3.4) shows whether the IoT device stays in some specific locations with severe radio channel degradations.

The utility of the IoT device depends on the queuing cost $\hat{W}^{(k)}$, the computation latency $\hat{T}^{(k)}$, and the energy consumption $\hat{E}^{(k)}$. Let ψ represent the loss to the IoT device due to the failure to carry out a computation task in time and ν be the queuing weight. Let β and μ denote the importance of energy saving and fast computation, respectively. The utility $u^{(k)}$ is estimated by

$$\begin{aligned} u^{(k)} =& R^{(k)} - \psi \mathbb{I}\left(b^{(k+1)} < 0\right) - \beta \hat{E}^{(k)} \\ & - \mu \hat{T}^{(k)} - \nu \hat{W}^{(k)}. \end{aligned} \tag{3.5}$$

The next channel power gain $h^{(k+1)}$ is estimated based on the channel model given by (3.1). As shown in Fig. 3.9, the state $\tilde{\mathbf{s}}^{(k)} = \left[C_1^{(k)}, \chi^{(k)}, h^{(k+1)}, \hat{\rho}^{(k)}, C_0^{(k)}, b^{(k)}\right]$ with a known transition probability h_{mn}, i.e.,

$$\Pr\left(\tilde{\mathbf{s}}^{(k)} \middle| \mathbf{s}^{(k)}, \mathbf{x}^{(k)}\right) = h_{mn}. \tag{3.6}$$

The IoT device estimates intermediate utility $u(\tilde{\mathbf{s}}^{(k)}, \mathbf{x}^{(k)})$ via (3.5). Based on the received offloading reports, the estimated energy consumption, and the computation latency, the IoT device obtains the next state $\mathbf{s}^{(k+1)}$. The Q-function $Q\left(\tilde{\mathbf{s}}, \mathbf{x}\right)$ is then updated based on the immediate state $\tilde{\mathbf{s}}^{(k)}$ and utility $u(\tilde{\mathbf{s}}^{(k)}, \mathbf{x}^{(k)})$ according to the iterated Bellman equation as follows:

$$\begin{aligned} Q\left(\tilde{\mathbf{s}}^{(k)}, \mathbf{x}^{(k)}\right) \leftarrow & (1-\alpha)\, Q\left(\tilde{\mathbf{s}}^{(k)}, \mathbf{x}^{(k)}\right) + \alpha \Bigg(u\left(\tilde{\mathbf{s}}^{(k)}, \mathbf{x}^{(k)}\right) \\ & + \gamma \max_{\mathbf{x}' \in \mathbf{A}} Q\left(\mathbf{s}^{(k+1)}, \mathbf{x}'\right) \Bigg), \end{aligned} \tag{3.7}$$

where the learning rate $\alpha \in (0, 1]$ weighs the current offloading experience and the discount factor $\gamma \in [0, 1]$ indicates the myopic view of the IoT device regarding the future reward. The quality function is then updated as follows:

$$Q(\mathbf{s}, \mathbf{x}) \leftarrow \sum_{\tilde{\mathbf{s}} \in \Lambda} \Pr\left(\tilde{\mathbf{s}} | \mathbf{s}, \mathbf{x}\right) Q\left(\tilde{\mathbf{s}}, \mathbf{x}\right). \tag{3.8}$$

The offloading experience $(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, u^{(k)}, \mathbf{s}^{(k+1)})$ is used to build the Dyna architecture and generate J simulated experiences in each time slot. More specifically, the model learning depends on an occurrence count vector of the next state denoted by Φ', which is updated by

$$\Phi'\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, \mathbf{s}^{(k+1)}\right) \leftarrow \Phi'\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, \mathbf{s}^{(k+1)}\right) + 1. \tag{3.9}$$

The occurrence count vector in the simulated experience denoted by Φ is updated in each real offloading experience by

$$\Phi\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}\right) \leftarrow \sum_{\mathbf{s}' \in \Lambda} \Phi'\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, \mathbf{s}'\right). \tag{3.10}$$

The transition probability to reach the state $\mathbf{s}^{(k+1)}$ from the state-action pair $\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}\right)$ is denoted by $\Pi\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, \mathbf{s}^{(k+1)}\right)$ and updated by

$$\Pi\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, \mathbf{s}^{(k+1)}\right) \leftarrow \frac{\Phi'\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, \mathbf{s}^{(k+1)}\right)}{\Phi\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}\right)}. \tag{3.11}$$

The reward record denoted by G' is the utility of the IoT device from the real offloading experience $u^{(k)}$, i.e.,

$$G'\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, \Phi\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}\right)\right) = u^{(k)}. \tag{3.12}$$

The average reward function over all the occurrence realizations denoted by G is updated by

$$G\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}\right) = \frac{1}{\Phi\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}\right)} \sum_{\kappa=1}^{\Phi\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}\right)} G'\left(\mathbf{s}^{(k)}, \mathbf{x}^{(k)}, \kappa\right). \tag{3.13}$$

The J simulated experiences are then generated from the Dyna architecture model (Π, G) via search control. Each simulated experience at time slot k leads to an additional Q-function update. More specifically, in the j-th update, the IoT device first randomly chooses a state-action pair $(\bar{\mathbf{s}}^{(j)}, \bar{\mathbf{x}}^{(j)})$ and selects the next state $\bar{\mathbf{s}}^{(j+1)}$ based on the state transition probability Π given by (3.11). The modeled reward $\hat{u}^{(j)}$

depends on the reward function G in (3.13) with the state-action pair $(\bar{\mathbf{s}}^{(j)}, \bar{\mathbf{x}}^{(j)})$ as follows:

$$\hat{u}^{(j)}\left(\bar{\mathbf{s}}^{(j)}, \bar{\mathbf{x}}^{(j)}\right) = G\left(\bar{\mathbf{s}}^{(j)}, \bar{\mathbf{x}}^{(j)}\right). \tag{3.14}$$

The Q-function for $(\bar{\mathbf{s}}^{(j)}, \bar{\mathbf{x}}^{(j)})$ is updated via the iterated Bellman equation again with (3.7).

As shown in Fig. 3.8, this scheme uses a transfer learning method named hotbooting as developed in [49] to initialize the Q-values with the computation offloading experiences in similar environments to save the initial random exploration. More specifically, the Q-values of the offloading algorithms after ξ offloading experiences are randomly selected from the offloading experience pool, which are denoted by $\overline{\mathbf{Q}}$ and used to initialize the Q-values in Algorithm 2.

3.5.2 *Performance Analysis*

We analyze the performance of the RL-based privacy-aware offloading scheme regarding the privacy level, energy consumption, computation latency, and utility. Similar to [4] and [50], we focus on the delay-sensitive applications, i.e., both the local execution and the task offloading can be performed within a time slot. For simplicity, both the computation latency of the edge device and the transmission latency of the computation results are assumed to be negligible, while this algorithm works in the other scenarios as well.

At time slot k, the IoT device offloads $x_1^{(k)}(C_1^{(k)} + C_0^{(k)})$ maritime data to the edge device, locally processes $x_0^{(k)}(C_1^{(k)} + C_0^{(k)})$ data, and stores the rest in the buffer to be processed in the future. According to [2, 51], the IoT device consumes $\varsigma x_0^{(k)}(C_1^{(k)} + C_0^{(k)})$ energy to compute the sensing data locally and takes $Px_1^{(k)}(C_1^{(k)} + C_0^{(k)})$ energy in the offloading process. Thus, we have

$$E = Px_1\left(C_1 + C_0\right) + \varsigma x_0\left(C_1 + C_0\right). \tag{3.15}$$

The data rate in the offloading can be modeled with $\log_2(1+Ph)$. The IoT device takes $T_0^{(k)} = x_0^{(k)}(C_1^{(k)} + C_0^{(k)})/f$ to compute the local sensing data at time slot k [51], and the resulting offloading latency is $T_1^{(k)} = x_1^{(k)}(C_1^{(k)} + C_0^{(k)})/\log_2(1 + Ph^{(k)})$. Thus, by (3.2), the total computation latency of the IoT device denoted by $T^{(k)}$ is given by

$$T = \max\left\{\frac{x_0\left(C_1 + C_0\right)}{f}, \frac{x_1\left(C_1 + C_0\right)}{\log_2\left(1 + Ph\right)}\right\}. \tag{3.16}$$

According to [52], the average queuing delay linearly increases with the average queue length and the priority of the generated maritime data $\chi^{(k)}$. Thus, the queuing cost denoted by $W^{(k)}$ is defined as

$$W = \chi\left(1 - x_0 - x_1\right)\left(C_1 + C_0\right). \tag{3.17}$$

The offloading selection over multiple time slots can be viewed as an MDP, as the future state is independent of the previous states for the given current state and offloading policy. Therefore, the RL-based offloading scheme in Algorithm 2 can achieve the optimal policy via trial and error without being aware of the privacy leakage, energy consumption, and edge computation model.

Theorem 3.1 *The maritime IoT device that uses Algorithm 2 in the dynamic offloading game can achieve the optimal policy given by* $\mathbf{x}^* = [0, 1]$*, and the privacy level is* C_0*. The computation latency, the energy consumption, and the utility are given respectively by*

$$T = \frac{C_1 + C_0}{\log_2\left(1 + Ph\right)} \tag{3.18}$$

$$E = P\left(C_1 + C_0\right) \tag{3.19}$$

$$u = C_0 - \left(\beta P + \frac{\mu}{\log_2\left(1 + Ph\right)}\right)\left(C_1 + C_0\right), \tag{3.20}$$

if

$$h > \hat{h} \tag{3.21}$$

$$\mu < \left(\nu\chi + 1 - \beta P\right)\log_2\left(1 + Ph\right) \tag{3.22}$$

$$\nu\chi < \beta\varsigma \tag{3.23}$$

$$b + \rho > P\left(C_1 + C_0\right). \tag{3.24}$$

Proof By (3.5), if (3.21) and $b + \rho - \varsigma\left(C_1 + C_0\right)x_0 > P\left(C_1 + C_0\right)x_1$, we have

$$\begin{aligned} u(\mathbf{x}) = &\left(\nu\chi - \beta P + 1 - \frac{\mu}{\log_2\left(1 + Ph\right)}\right)\left(C_1 + C_0\right)x_1 \\ &+ \left(\nu\chi - \beta\varsigma\right)\left(C_1 + C_0\right)x_0 - C_1 - \nu\chi\left(C_1 + C_0\right) \\ &- \psi\mathbb{I}\left(b + \rho - \varsigma\left(C_1 + C_0\right)x_0 - P\left(C_1 + C_0\right)x_1\right) \\ = &\left(\nu\chi - \beta P + 1 - \frac{\mu}{\log_2\left(1 + Ph\right)}\right)\left(C_1 + C_0\right)x_1 \\ &+ \left(\nu\chi - \beta\varsigma\right)\left(C_1 + C_0\right)x_0 - C_1 - \nu\chi\left(C_1 + C_0\right). \end{aligned} \tag{3.25}$$

If (3.22)–(3.23), $\forall\, \mathbf{x} \in \mathbf{A}$,

$$\frac{\partial u}{\partial x_0} = (\nu\chi - \beta\varsigma)(C_0 + C_1) < 0, \tag{3.26}$$

indicating that the utility decreases with x_0.

$$\frac{\partial u}{\partial x_1} = \left(\nu\chi - \beta P + 1 - \frac{\mu}{\log_2(1 + Ph)}\right)(C_0 + C_1) > 0, \tag{3.27}$$

indicating that the utility with x_1. As $x_0 \in [0, 1]$ and $x_1 \in [0, 1]$, we have $\arg\max_{\mathbf{x}\in\mathbf{A}} u = [0, 1]$.

According to [42], this RL-based scheme can achieve the optimal policy $\mathbf{x}^* = [0, 1]$ in the MDP after a sufficiently long time. Therefore, this algorithm can achieve $\mathbf{x}^* = [0, 1]$. If (3.24), by (3.4), (3.16) and (3.15), we have $R = C_0$ and prove (3.18)–(3.20).

Remark 1 A maritime IoT device applies the RL-based privacy-aware offloading algorithm to achieve the optimal policy without being aware of the privacy leakage, energy consumption, and edge computing model in the dynamic offloading process. If the IoT device has a good radio channel to the edge device as shown in (3.22), the local processing energy overhead is high as shown in (3.23), and the offloading energy consumption is low as shown in (3.24), and the IoT device will offload all the computation tasks to the edge device. In this case, the privacy level of the IoT device equals the size of the buffered tasks, and both the computation latency and energy consumption increase linearly with the size of the total computation tasks as indicated in (3.18) and (3.19).

Theorem 3.2 *The maritime IoT device using Algorithm 2 in the dynamic offloading game can achieve the optimal policy given by* $\mathbf{x}^* = [1, 0]$*, and the privacy level is* C_1*. The computation latency, the energy consumption, and the utility are given respectively by*

$$T = \frac{C_1 + C_0}{f} \tag{3.28}$$

$$E = \varsigma(C_1 + C_0) \tag{3.29}$$

$$u = C_1 - \left(\beta\varsigma + \frac{\mu}{f}\right)(C_1 + C_0), \tag{3.30}$$

if

$$h > \hat{h} \tag{3.31}$$

$$\nu\chi < \beta P + 1 \tag{3.32}$$

$$\mu < f(\nu\chi - \beta\varsigma) \tag{3.33}$$

$$b + \rho > \varsigma(C_1 + C_0). \tag{3.34}$$

Proof The proof is similar to that of Theorem 3.1.

Remark 2 If the maritime sensing data seem normal as shown in (3.32), the offloading energy overhead is high, as shown in (3.33), and the IoT device has powerful computation resources, as shown in (3.34), and the IoT device will process all the computation tasks locally. In this case, the privacy level equals the size of the new sensing data, and both the IoT energy consumption and the computation latency increase with the total computation task size as indicated in (3.28) and (3.29).

Theorem 3.3 *The maritime IoT device using Algorithm 2 in the dynamic offloading game can achieve the optimal policy given by* $\mathbf{x}^* = [0, C_0/(C_1 + C_0)]$*, and the privacy level is* ω*. The computation latency, the energy consumption, and the utility are given respectively by*

$$T = \frac{C_0}{\log_2(1 + Ph)} \tag{3.35}$$

$$E = PC_0 \tag{3.36}$$

$$u = \omega - \psi - \left(\beta P - \frac{\mu}{\log_2(1 + Ph)}\right) C_0, \tag{3.37}$$

if

$$h < \check{h} \tag{3.38}$$

$$\mu < (\nu\chi - \beta P)\log_2(1 + Ph) \tag{3.39}$$

$$\nu\chi < \beta\varsigma \tag{3.40}$$

$$b + \rho < PC_0. \tag{3.41}$$

Proof The proof is similar to that of Theorem 3.1.

Remark 3 If the offloading energy overhead is low as shown in (3.39), the local processing energy overhead is high as shown in (3.40), and the IoT device has insufficient computation resources as shown in (3.41); thus, the IoT device will offload some sensing data to the edge device and store the rest of the tasks to the buffer to protect the user privacy. In this case, the IoT device can achieve a privacy level given by ω, and both the computation latency and the energy consumption increase linearly with the size of the buffered tasks as indicated in (3.35) and (3.36).

3.6 Simulation Results

Simulations have been performed to evaluate the RL-based privacy-aware offloading scheme in dynamic maritime IoT communication systems. In the simulations, each time slot lasts 1 s, and the maritime IoT device generates new maritime data of 30 kb. According to [4], the IoT device consumes 10^{-4} J energy to locally process one bit sensing data and uses 0.2 J energy to send one bit sensing data to the edge device. The queuing delay, the location privacy over usage pattern privacy, the energy consumption, and the computation latency are weighted with 40, 5, 2.5, and 5, respectively. If not specified otherwise, the learning rate is 0.8, the discount factor is 0.7, and ϵ is 0.1 according to [53]. The CMDP-based offloading scheme in [20] has been evaluated in the simulations as a benchmark.

As shown in Fig. 3.10, the RL-based offloading scheme converges to the performance bound given by Theorem 3.2. This scheme exceeds the CMDP-based offloading scheme as proposed in [20] with a higher privacy level. This scheme also saves the energy consumption of the IoT device, reduces the computation latency, and increases the utility of the IoT device. For instance, this scheme improves 36.63% of the privacy level, saves 9.63% of the energy consumption, and decreases

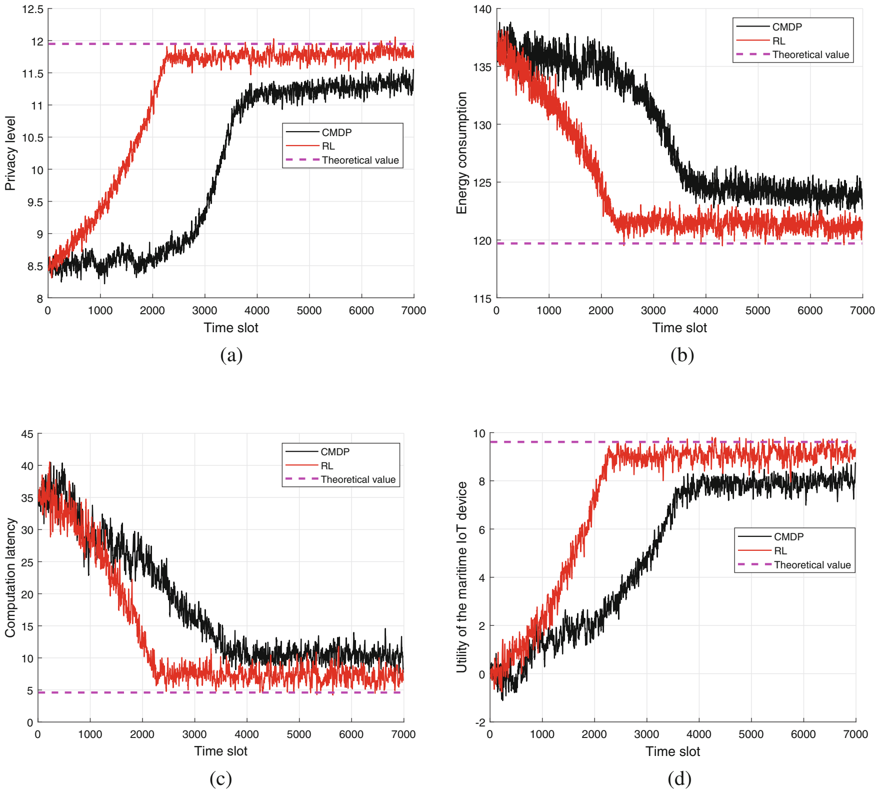
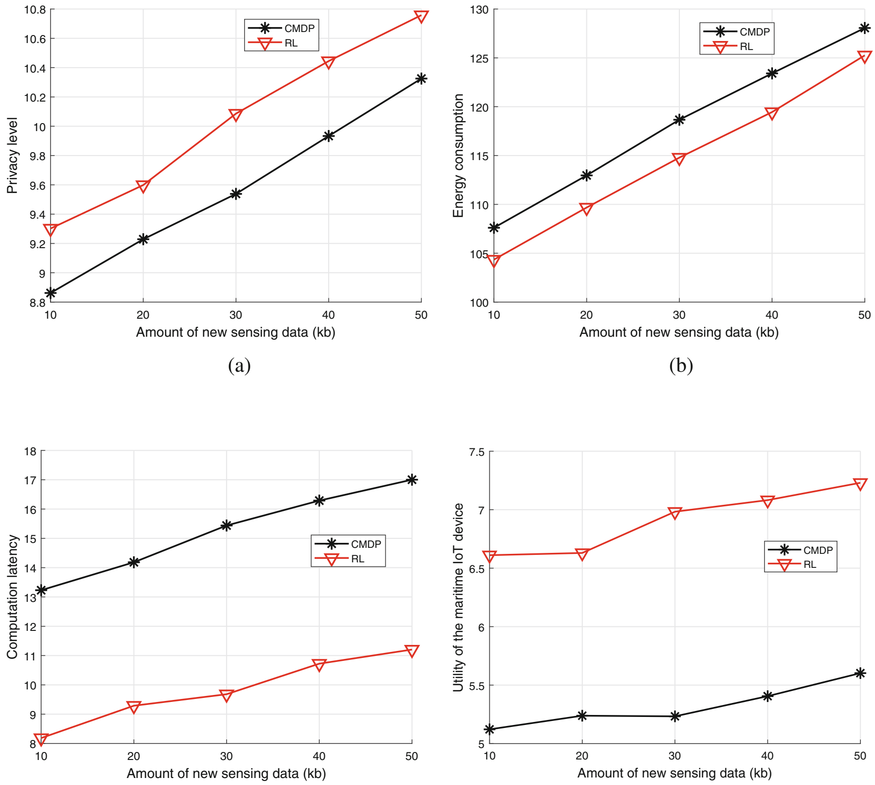


Fig. 3.10 Performance of the privacy-aware offloading scheme in a maritime IoT device with EH. (**a**) Achieved privacy level. (**b**) Energy consumption of the maritime IoT device. (**c**) Computation latency. (**d**) Utility of the maritime IoT device



(a)

(b)

(c)

(d)

Fig. 3.11 Performance of the privacy-aware offloading scheme in a maritime IoT device with EH that generates an amount of the sensing data in each time slot. (**a**) Achieved privacy level. (**b**) Energy consumption of the maritime IoT device. (**c**) Computation latency. (**d**) Utility of the maritime IoT device

68.79% of the computation latency, compared with the CMDP-based scheme at the 2200-th time slot. Consequently, as shown in Fig. 3.10d, the utility of the maritime IoT device increases about two times compared with that of the CMDP-based scheme. Figure 3.10 shows that the RL-based scheme accelerates the learning speed, e.g., this scheme saves 40% of time slots to reach the privacy level of 11 compared with the CMDP. This is due to the fact that the transfer learning technique, a PDS method, and a Dyna architecture are used to accelerate the learning speed of the maritime IoT device with the extended state space.

The offloading performance averaged over the first 4500 time slots in the dynamic offloading game is shown in Fig. 3.11. In the simulations, an IoT device has to compute 10 to 50 kb new maritime sensing data in each time slot. The privacy level of the IoT device increases, as the amount of the sensing data changes from 10 kb to 50 kb. For instance, if the ship has to process 50 kb maritime data instead of 10 kb maritime data, the privacy level, the energy consumption, the computation latency, and the utility of the maritime IoT device with RL-based offloading increase by 28.93%, 40.78%, 1 time, and 28.79%, respectively. If the maritime IoT device has to process 50 kb maritime data in each time slot as shown in Fig. 3.11, the RL-based offloading scheme exceeds the benchmark CMDP scheme with 12.39% higher privacy level, 5.38% lower energy consumption, and 32.35% shorter computation latency.

3.7 Conclusion

In this chapter, we have presented an RL-based privacy-aware offloading scheme for an EH-powered maritime IoT device to choose the offloading rate and the local computing rate without being aware of the privacy leakage, IoT energy consumption, and edge computation model. This scheme evaluates the privacy level, the energy consumption, and the computation latency to choose the offloading policy to the edge device in each time slot. The RL-based offloading scheme uses the transfer learning technique, a known radio channel model, and a Dyna architecture to accelerate the learning speed for dynamic maritime IoT communication systems. We prove that this scheme can achieve the optimal offloading policy in the dynamic offloading process and provide its performance upper bound in terms of the privacy level, the computation latency, and the energy consumption. Simulation results show that this scheme improves the privacy level by 36.63%, saves 9.63% of the energy consumption, and decreases 68.79% of the computation latency compared with the benchmark CMDP scheme.

References

1. Y. Mao, C. You, J. Zhang, et al., A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tut. **19**(4), 2322–2358 (2017)
2. W. Zhang, Y. Wen, K. Guan, et al., Energy-optimal mobile cloud computing under stochastic wireless channel. IEEE Trans. Wirel. Commun. **12**(9), 4569–4581 (2013)
3. C. You, K. Huang, H. Chae, et al., Energy-efficient resource allocation for mobile-edge computation offloading. IEEE Trans. Wirel. Commun. **16**(3), 1397–1411 (2017)
4. Y. Mao, J. Zhang, K. Letaief, Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE J. Sel. Areas Commun. **34**(12), 3590–3605 (2016)
5. P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, T. Taleb, Survey on multi-access edge computing for internet of things realization. IEEE Commun. Surv. Tut. **20**(4), 2961–2991 (2018)
6. T.Q. Dinh, J. Tang, Q.D. La, T.Q.S. Quek, Offloading in mobile edge computing: task allocation and computational frequency scaling. IEEE Trans. Commun. **65**(8), 3571–3584 (2017)
7. T. Yang, H. Feng, Gao, et al., Two-stage offloading optimization for energy–latency tradeoff with mobile edge computing in maritime internet of things. IEEE Internet Things J. **7**(7), 5954–5963 (2020)

8. S. Jiang, X. Su, Y. Zhou, Data privacy protection for maritime mobile terminals, in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP)*, Changsha, China, Oct. 2021
9. S. Ulukus, A. Yener, E. Erkip, et al., Energy harvesting wireless communications: a review of recent advances. IEEE J. Sel. Areas Commun. **33**(3), 360–381 (2015)
10. J.W. Kimball, B.T. Kuhn, R.S. Balog, A system design approach for unattended solar energy harvesting supply. IEEE Trans. Power Electron. **24**(4), 952–962 (2009)
11. M. Han, J. Duan, S. Khairy, L.X. Cai, Enabling sustainable underwater iot networks with energy harvesting: a decentralized reinforcement learning approach. IEEE Internet Things J. **7**(10), 9953–9964 (2020)
12. F. Kong, C. Dong, X. Liu, H. Zeng, Quantity versus quality: optimal harvesting wind power for the smart grid. Proc. IEEE **102**(11), 1762–1776 (2014)
13. S. Kim, R. Vyas, Bito, et al., Ambient RF energy-harvesting technologies for self-sustainable standalone wireless sensor platforms. Proc. IEEE **102**(11), 1649–1666 (2014)
14. Y. Sun, C. Song, S. Yu, et al., Energy-efficient task offloading based on differential evolution in edge computing system with energy harvesting. IEEE Access **9**, 16,383–16,391 (2021)
15. J. Xu, L. Chen, S. Ren, Online learning for offloading and autoscaling in energy harvesting mobile edge computing. IEEE Trans. Cogn. Commun. Netw. **3**(3), 361–373 (2017)
16. F. Wang, J. Xu, X. Wang, et al., Joint offloading and computing optimization in wireless powered mobile-edge computing systems. IEEE Trans. Wirel. Commun. **17**(3), 1784–1797 (2017)
17. A. Hosseini-Fahraji, P. Loghmannia, K. Zeng, et al., Energy harvesting long-range marine communication, in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Toronto, CanadaJul, Jul. 2020
18. Y. Chao, Autonomous underwater vehicles and sensors powered by ocean thermal energy, in *Proceedings of the OCEANS*, Shanghai, China, Apr. 2016
19. A. Alrawais, A. Alhothaily, C. Hu, et al., Fog computing for the internet of things: security and privacy issues. IEEE Internet Comput. **21**(2), 34–42 (2017)
20. X. He, J. Liu, R. Jin, et al., Privacy-aware offloading in mobile-edge computing, in *Proceedings of the IEEE Global Communications Conference (GlobeCom)*, Singapore, Dec. 2017
21. L. Xiao, X. Wan, C. Dai, et al., Security in mobile edge caching with reinforcement learning. IEEE Wirel. Commun. **25**(3), 1–7 (2018)
22. S. Yi, Z. Qin, Q. Li, Security and privacy issues of fog computing: a survey, in *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications (WASA)*, Qufu, China, Aug. 2015
23. J. Ni, A. Zhang, X. Lin, et al., Security, privacy, and fairness in fog-based vehicular crowdsensing. IEEE Commun. Mag. **55**(6), 146–152 (Jun. 2017)
24. C. Xu, J. Ren, Y. Zhang, et al., Dppro: differentially private high-dimensional data release via random projection. IEEE Trans. Inf. Forensic Secur. **12**(12), 3081–3093 (2017)
25. J. Liu, K. Kumar, Y. Lu, Tradeoff between energy savings and privacy protection in computation offloading, in *Proceedings of the ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED)*, Austin, TX, Aug. 2010
26. J. Liu, Y. Lu, Energy savings in privacy-preserving computation offloading with protection by homomorphic encryption, in *Proceedings of the International Conference Power-Aware Computing and Systems (HotPower)*, Berkeley, CA, Oct. 2010
27. R. Lu, X. Lin, X. Shen, SPOC: a secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency. IEEE Trans. Parallel. Distrib. Syst. **24**(3), 614–624 (2013)
28. Y. Wang, W. Feng, J. Wang, T.Q. Quek, Hybrid satellite-UAV-terrestrial networks for 6G ubiquitous coverage: a maritime communications perspective. IEEE J. Sel. Areas. Commun. **39**(11), 3475–3490 (2021)
29. H. Heidari, O. Onireti, R. Das, M. Imran, Energy harvesting and power management for IoT devices in the 5G era. IEEE Commun. Mag. **59**(9), 91–97 (2021)
30. T. Yang, H. Feng, C. Yang, et al., Multivessel computation offloading in maritime mobile edge computing network. IEEE Internet Things J. **6**(3), 4063–4073 (2018)

31. N. Abbas, Y. Zhang, A. Taherkordi, et al., Mobile edge computing: a survey. IEEE Internet Things J. **5**(1), 450–465 (2018)
32. M. Chiang, T. Zhang, Fog and IoT: an overview of research opportunities. IEEE Internet Things J. **3**(6), 854–864 (2016)
33. J. Ren, H. Guo, C. Xu, et al., Serving at the edge: a scalable IoT architecture based on transparent computing. IEEE Netw. **31**(5), 96–105 (2017)
34. X. Peng, J. Ren, L. She, et al., BOAT: a block-streaming app execution scheme for lightweight iot devices. IEEE Internet Things J. **5**(3), 1816–1829 (2018)
35. S. Gao, T. Yang, H. Ni, G. Zhang, Multi-armed bandits scheme for tasks offloading in mec-enabled maritime communication networks, in *Proceedings of the IEEE/CIC International Conference on Communications in China (ICCC)*, Chongqing, China, Aug. 2020
36. T. Xia, M.M. Wang, J. Zhang, L. Wang, Maritime internet of things: challenges and solutions. IEEE Wirel. Commun. **27**(2), 188–196 (2020)
37. J. Pavur, D. Moser, M. Strohmeier, V. Lenders, I. Martinovic, A tale of sea and sky on the security of maritime vsat communications, in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, Jul. 2020
38. X. Chen, J. Wu, Y. Cai, et al., Energy-efficiency oriented traffic offloading in wireless networks: a brief survey and a learning approach for heterogeneous cellular networks. IEEE J. Sel. Areas Commun. **33**(4), 627–640 (2015)
39. L. Xiao, C. Xie, T. Chen, et al., A mobile offloading game against smart attacks. IEEE Access **4**, 2281–2291 (2016)
40. C. Li, J. Xia, F. Liu, D. Li, L. Fan, G.K. Karagiannidis, A. Nallanathan, Dynamic offloading for multiuser muti-cap mec networks: a deep reinforcement learning approach. IEEE Trans. Veh. Technol. **70**(3), 2922–2927 (2021)
41. Y.-C. Wu, T.Q. Dinh, Y. Fu, C. Lin, T.Q.S. Quek, A hybrid DQN and optimization approach for strategy and resource allocation in MEC networks. IEEE Trans. Wirel. Commun. **20**(7), 4282–4295 (2021)
42. R. Sutton, A. Barto, *Reinforcement Learning: An Introduction* (MIT press, Cambridge, MA, 1998)
43. X. He, H. Dai, P. Ning, Improving learning and adaptation in security games by exploiting information asymmetry, in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Hong Kong, China, Aug. 2015
44. S.J. Pan, Q. Yang, A survey on transfer learning. IEEE Trans. Knowl. Data. Eng. **22**(10), 1345–1359 (2010)
45. W. Liu, J. Cao, L. Yang, et al., Appbooster: boosting the performance of interactive mobile applications with computation offloading and parameter tuning. IEEE Trans. Parallel. Distrib. Syst. **28**(6), 1593–1606 (2017)
46. N. Golrezaei, A. Molisch, A. Dimakis, et al., Femtocaching and device-to-device collaboration: a new architecture for wireless video distribution. IEEE Commun. Mag. **51**(4), 142–149 (2013)
47. M. Xia, S. Aissa, On the efficiency of far-field wireless power transfer. IEEE Trans. Signal Proces. **63**(11), 2835–2847 (2015)
48. D. Zhang, Y. Qiao, L. She, et al., Two time-scale resource management for green internet of things networks. IEEE Internet Things J. **6**(1), 545–556 (2019)
49. L. Xiao, Y. Li, C. Dai, et al., Reinforcement learning-based NOMA power allocation in the presence of smart jamming. IEEE Trans. Veh. Technol. **67**(4), 3377–3389 (2018)
50. O. Munoz, A. Pascual-Iserte, J. Vidal, Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading. IEEE Trans. Veh. Technol. **64**(10), 4738–4755 (2015)
51. X. Chen, L. Jiao, W. Li, et al., Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE Trans. Netw. **24**(5), 2795–2808 (2016)
52. N. Mastronarde, M. Van der Schaar, Fast reinforcement learning for energy-efficient wireless communication. IEEE Trans. Signal Proces. **59**(12), 6262–6266 (2011)
53. L. Xiao, Y. Li, X. Huang, et al., Cloud-based malware detection game for mobile devices with offloading. IEEE Trans. Mob. Comput. **16**(10), 2742–2750 (2017)

Chapter 4
Learning-Based Resource Management for Maritime Communications

With the rapid development of smart maritime services, more and more underwater vehicles, ships, sensors and underwater industrial the 5G networks need to support interconnect of a large number of smart maritime communication devices [1–3], which make decisions independently or collaboratively based on reinforcement learning techniques [3]. However, the learning-based wireless networks that support maritime applications in smart ocean, intelligent transportation, automatic industry, meter auto reporting, and remote sensing give rise to key challenges.

First, the large amount of data generated by massive smart devices raises challenges of collecting, integrating, storing, accessing, and processing data, as well as data mining for the behavior and characteristic discovery of wireless maritime networks [3–5].

Second, due to the extremely long range of service requirements of maritime devices and the complex/dynamic communication environments, existing wireless communication systems are still not smart enough to tackle optimized physical layer designs, sophisticated learning, complicated decision-making, and efficient resource management tasks [1]. To fulfill the potential benefits of maritime networks and deal with the growing challenges, recent research on machine learning has drawn attentions as a promising solution.

Machine learning, as the most powerful artificial intelligence technique, has already been widely applied in computer vision, signal/language processing, social behavior analysis, projection management, and so on [6]. Explicitly, machine learning methods analyze observations/data/experience to find the patterns and underlying structures and enable machines/systems to learn automatically without human intervention and adjust actions accordingly.

Machine learning mainly consists of three categories: supervised learning, unsupervised learning, and reinforcement learning [6]. Supervised learning algorithms depend on labeled training samples, while unsupervised learning algorithms do not rely on the labels of the training data to provide inference services and reinforcement learning that has attracted extensive research attentions in the field of wireless

L. Xiao et al., *Reinforcement Learning for Maritime Communications*, Wireless Networks, https://doi.org/10.1007/978-3-031-32138-2_4

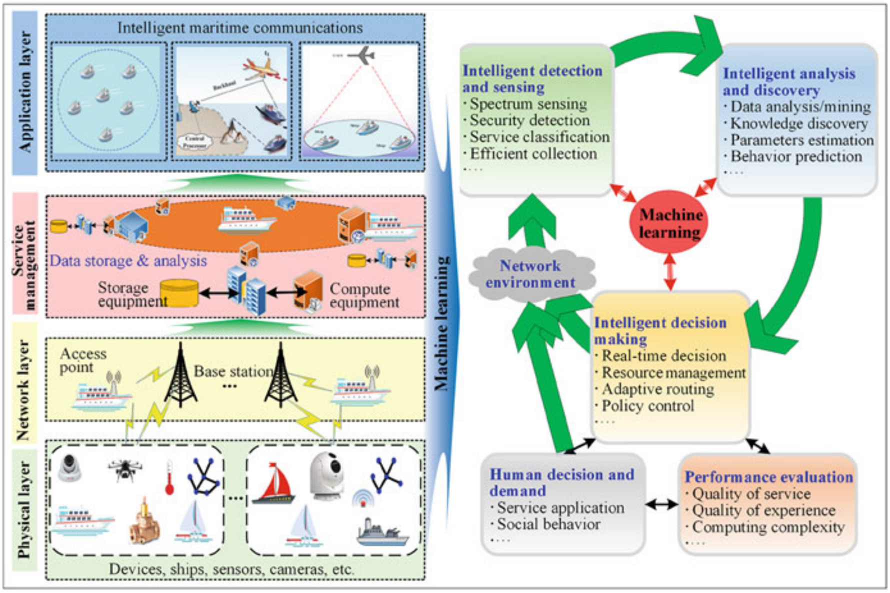


Fig. 4.1 Functional diagram of intelligent maritime communications

communications and security and enables communication devices to learn how to map situations to actions to maximize a reward by interacting with the network environment.

As shown in Fig. 4.1, thanks to machine learning techniques, intelligent wireless maritime communication networks are capable of tackling the detection and sensing tasks (e.g., robust detection and efficient data collection), data analysis and discovery tasks (e.g., knowledge discovery and behavior prediction), as well as decision-making tasks (e.g., resource management and policy control) from the physical layer to the application layer.

In particular, machine learning offers a versatile set of algorithms to analyze numerous data/observations and discover the depth knowledge. This effectively assists cognitive wireless networks to adapt network protocols and decision-making for different services in different communication scenarios and solve various technical problems, such as signal processing, parameter optimization, behavior analysis, mobile management, and resource management [1], [4], [5], [7–15]. However, how to design machine learning algorithms to address the above problems in maritime wireless networks remains a significant challenge.

In this chapter, we consider how to apply machine learning, such as reinforcement learning techniques, to address the challenges mentioned above in maritime communication networks. Table 4.1 presents the family tree of the three categories of machine learning (i.e., supervised, unsupervised, and reinforcement learning) and their potential applications in maritime communication networks. We review basic

Table 4.1 Machine learning for maritime communications

Category	Tasks	Algorithms	Applications and references
Supervised learning	Classification	Support vector machine, K-nearest neighbors	Security/interference detection, image/service behavior classification, spectrum sensing
	Regression	Linear regression, support vector regression, Gaussian process regression	Channel estimation, mobility prediction, cross-layer handover
Unsupervised learning	Clustering	K-means clustering, neural network	Device clustering, filtering designs, localization, service segmentation
	Dimension reduction	Principal component analysis, isometric mapping	Big data visualization, interference filtering, data compression, feature elicitation
Reinforcement learning	Policy/value iteration learning	Markov decision process, Q-learning, policy gradient, actor critic, deep Q-network,	Decision-making, packet transmission, spectrum access, network association, energy harvesting/efficiency, adaptive routing, resource management

algorithms in each category and discuss typical examples on how to apply such algorithms to improve the performance of wireless maritime networks.

4.1 Reinforcement Learning Principle

In particular, reinforcement learning enables a learning agent to choose the actions via trial and errors in the MDP even in dynamic environments, to maximize the long-term expected reward. Algorithms. For example, MDP, Q-learning, policy gradient, actor critic (AC), and DRL have been applied to improve wireless communication efficiency and security [6].

MDP Models for Network Association and Vehicular Routing Models In the Markov decision process, in which the future states are independent of the previous states given the action taken in the current state, and the RL is promising to yield, the optimal policy based on the interaction with the environment in discrete time steps. At each time k, the agent takes action $\mathbf{a}$ from the current state $\mathbf{s^k}$ to a new state $\mathbf{s}^{k+1}$ and calculates the corresponding reward U^k. During this process, the probability of moving from the current state to a new state is described by the transition probability P. The learning agent evaluates the quality of the action $\mathbf{a}$ based on the immediate reward $\mathbf{u}$ as well as the cumulative reward to explore the optimal policy in the next time step. In MDP, if a learning agent only has partial knowledge of the environment state or the reward of each action, possibly due to the limited feedback from the environment and state estimation errors, the learning problem is viewed as a partially observable Markov decision process (POMDP), which often corresponds to severe learning performance degradation.

Applications In cognitive wireless networks, the MDP or POMDP model has been applied in [4], [10], [11], [14] to improve the spectrum access, network association, energy harvesting, and load-balancing, in which the smart communication devices as the learning agents interact with the network that constitutes the environment. For example, in cognitive wireless networks, a large amount of wireless smart devices always choose the evolved NodeB (eNB) with the best signal quality for the attachment, thereby leading to serious network congestion and overload. In the eNB selection problem, the fixed bandwidth of eNB, the limited energy of devices, and the time-variant channels define environment features, and the devices' connection, their transmission power, and the transmit packet size can be regarded as the actions in each time slot.

Due to uncertainties in the state of mobile devices and their actions' effect to the state dynamics, such as the partial observation of the environment and imperfect position tracking/navigation, the decision-making problem can be formulated as a POMDP model under the partial knowledge. For instance, in large-scale vehicular cognitive wireless networks, the traffic situations are highly complicated, uncertain, dynamic, and only partially observable. Hence, the decision-making problem (e.g., automated driving, adaptive routing) can be formulated as a POMDP model, which can effectively avoid collisions and decrease the traffic congestion and enhance the driving safety and vehicle-network resource utilization efficiency [11].

Value/Policy-Iteration Learning Models for Policy Control and Resource Allocation Models Existing RL algorithms can be approximately divided into two groups: value iteration and policy iteration. Value iteration, e.g., Q-learning, starts with a random value function and then iteratively updates the value function until achieving the optimal value function. The best policy can be derived based on the optimal value function. In contrast, policy iteration, e.g., policy gradient, randomly selects a starting policy and iterates towards the optimal policy until the policy converges based on the value function evaluation [6]. Q-learning, as a model-free and value-iteration RL algorithm, solves the MDP problem even in dynamic and unknown environment models. The learning agent in the Q-learning model uses a Q-function to estimate its accumulated reward.

In policy gradients (PG), the learning agent seeks to directly optimize the policy function instead of the Q-function in Q-learning in the policy space. The optimization is derived directly by maximizing the expected reward. As another important RL algorithm, AC combines the benefits of the value-iteration and policy-iteration models and designs both the actor and the critic network. More specifically, the actor is represented by adopting a control policy with action selections. The critic evaluates the input policy by a reward function. In addition, DRL applies the deep learning techniques, e.g., deep neural networks, to compress the high-dimensional state space observed by the learning agent and directly use the deep learning network to represent the value function or policy model, such as the deep Q-learning and deep Q-network.

Applications The RL algorithms have been widely applied in large-scale cognitive wireless networks to support the intelligent decision-making for resource management, channel access, interference coordination, transmission scheduling, power control, and so on [1], [7–11], [14], [15]. The RL algorithms enable the network to optimize the policies independently among devices with the minimal human interaction. For example, a large number of smart devices entail a significant increase in the energy consumption in cognitive wireless networks, and thus the energy minimization problem becomes more challenging. Hence, the scheduling framework incorporating RL enables the scheduler to intelligently develop an association between the optimal action and the current state of the environment to minimize the energy consumption with variable workloads.

Conventional RL algorithms such as Q-learning and AC are suitable to make decisions with handcrafted features or low-dimensional data. On the other hand, DRL enables communication devices to learn their action-value policies directly from complex high-dimensional states. For instance, in dynamic networks, the channel conditions, ship requirements, and cache storage are all time-varying, and the network has a large number of environment states, e.g., device status (sleep or active), channel quality, and channel status (busy or idle). The DRL-based resource allocation uses deep learning to estimate the reward of each feasible policy for the devices with sufficient computing resources for faster learning speed [14].

Motivated by the above analysis and observations, in order to address the abovementioned challenges in massive access for 5G and B5G wireless networks, this chapter not only studies how to manage the massive access requests from a huge number of devices but also takes various QoS requirements (ranging from strict low latency and high reliability to minimum data rate) into consideration. Besides, a novel distributed cooperative learning approach-based QoS-aware massive access is presented to optimize the joint subchannel assignment and transmission power control strategy without a centralized controller. The main contributions of the chapter are summarized as follows:

- We formulate a joint subchannel assignment and transmission power control problem for massive access considering different practical QoS requirements, and the energy-efficient massive access management problem is modeled as a multi-agent RL problem. Hence, each ship has the ability to intelligently make its spectrum access decision according to its own instantaneous observations.

- A distributed cooperative subchannel assignment and transmission power control approach based on DRL is proposed for the first time to guarantee both the strict reliability and latency requirements on ultrareliable and low-latency communication (URLLC) services in a massive access scenario, where the latency constraint is transformed into a data rate constraint which can make the optimization problem tractable. Specifically, a proper QoS-aware reward function is built to cover both the network EE and devices' QoS requirements in the learning process.
- In addition, we apply transfer learning and cooperative learning mechanisms to enable communication links to work cooperatively in a distributed cooperative manner, in order to improve the network performance and transmission success probability based on local observation information. In detail, in transfer learning, if a new ship joins the network or applies a new service, or one communication link achieves poor performance (e.g., low QoS satisfaction level or low convergence speed), then it can directly search the expert agent from the neighbors and utilize the transfer learning model from the expert agent instead of building a new learning model. In cooperative learning, ships are encouraged to share their selected actions with their neighbors and take turns to make decisions, which can enhance the overall benefit by choosing the actions jointly instead of independently.
- Extensive simulation results are presented to verify the effectiveness of the proposed distributed cooperative learning approach in massive access scenario and demonstrate the superiority of the proposed learning approach in terms of meeting the network EE and improving the transmission success probability compared with other existing approaches.

The rest of this chapter is organized as follows. The related work is provided in Sect. 4.2. In Sect. 4.3, the system model and problem formulation are provided. The massive access management problem is modeled as a Markov decision-making process in Sect. 4.4. Section 4.5 proposes a distributed cooperative multi-agent learning-based massive access approach. Section 4.6 provides simulation results, and Sect. 4.7 concludes the chapter.

4.2 Related Work

With the rapid development of the IoTs, IoT devices access maritime wireless networks to support diverse applications, e.g., smart ocean animal tracking systems [1] and 5G and beyond 5G (B5G) networks are required to provide seamless access for maritime IoT devices with various services.

URLLC is one of the most challenging services with stringent low-latency and high-reliability requirements. For example, a general URLLC requirement of a one-way radio is 99.999% target reliability with 1 ms latency [4]. Consequently, URLLC entails great difficulty in massive access in 5G and B5G wireless networks, especially in maritime wireless communications.

To relieve the network congestion resulting from radio spectrum access in maritime communication, the hierarchical contention-based access model is proposed in [5] to enhance the access success probability and meet the QoS requirements of maritime devices. In [6], a time-division-multiple-access transmission protocol was presented to deal with the congestion caused by video packet transmission in maritime communications. Liu et al. in [8] investigated a priority-based multiple access protocol to ensure the data delivery for the given limit of the energy consumption and delay.

Furthermore, a relay-aided random access scheme as proposed in [9] provides IoT device access to the smart ocean. Interestingly, a distributed D2D resource allocation for maritime communication was designed to optimize the general energy efficiency of the network [10].

Besides, several methods were presented to enhance the traditional random access performance, such as access class barring (ACB), slotted access, and backoff [11], [12]. For instance, in [12], an efficient random access procedure based on ACB was investigated to decrease the access delay and the power consumption for the large-scale wireless networks with massive access.

The mentioned spectrum access approaches in [5–12] are simple, flexible, and able to support massive wireless connections without relying on central coordinator. However, the high transmission success probability is not easily guaranteed for URLLC applications.

To satisfy the critical requirements of URLLC in massive IoT or mMTC, recent studies have presented spectrum access solutions in [13–19]. For instance, Weerasinghe et al. proposed a priority-based massive access approach to support reliable and low-latency access for mMTC devices [13], where devices are categorized into a number of groups with different priority access levels. A probability density function of signal-to-noise ratio (SNR) was derived for a large number of uplink URLLC devices in [14], and numerical results verified that the presented model can satisfy the critical requirements of URLLC. Popovski et al. in [15] discussed the principles of wireless access for URLLC and provided a perspective on the relationship between latency, packet size, and bandwidth. In [16] and [17], grant-free spectrum access was adopted to reduce transmission latency and improve spectrum utilization in URLLC scenarios. In [18] and [19], different resource management schemes were developed to show how to update the system parameters that meet the URLLC requirements in industrial IoT networks, since industrial automation requires strict low latency and high reliability for manufacturing control.

Nevertheless, very few literatures such as [13] and [14] investigated how to meet strict URRLC requirements in massive access scenarios. Moreover, even in [13] and [14], the optimization objective is a single time slot optimization, and the massive access decision approaches sometimes converge to the suboptimal solution.

To address the massive access management problem in maritime communications, and support the stringent reliability and latency constraints, emerging technologies of 5G, i.e., massive MIMO, non-orthogonal multiple access (NOMA), and D2D communications, support massive connectivity over limited available radio resources. For example, Chen et al. in [12] and [20] presented non-orthogonal communication frameworks based on massive NOMA to support massive connections, and the transmit power values were optimized to mitigate severe co-channel interference by using interference cancellation techniques [21]. In addition, an application-specific NOMA-based communication architecture was investigated for future URLLC Tactile Internet [22].

In [14], [15], [23], and [24], coordinated and uncoordinated access protocols support massive connectivity in massive MIMO systems by exploiting large spatial degrees of freedom to admit massive IoT devices. Specifically, in [14] and [15], massive MIMO can be acted as a natural enabler for URLLC to support high capacity, spatial multiplexing, and diversity links. Moreover, a potential solution for the massive access is to offload a large amount of traffic onto D2D communication links [25], [26], to reduce the energy consumption and transmission delay, and improve spectrum efficiency. D2D-based URLLC transmission protocols as proposed in [27] and [28] classify the communication devices into groups based on their QoS requirements, i.e., stringent low-latency and high-reliability requirements, and allocate the radio resources accordingly.

In addition, EE plays an important role in green wireless networks. The reasons are that most of devices (e.g., sensors, actuators, and wearable devices) are power constrained and energy consumption is massive and expensive under high-density scenario of devices. In [29] and [30], the authors optimized the joint radio access and power resource allocation to maximize EE while guaranteeing the transmission delay requirements and transmit power constraints of a huge number of devices. To mitigate co-channel interference and further enhance the EE performance of NOMA-based systems with massive IoT devices, subchannel allocation and power control approaches were proposed in [19], [29], [31]. Furthermore, Miao et al. [32] proposed an energy-efficient clustering scheme to address spectrum access problem for massive M2M communications. Although the authors in [29–32] mainly focused on the EE maximization-based massive access, the different QoS requirements (such as latency and reliability) of devices have not been well studied in massive access scenario.

Considering that intelligence is an important characteristic of future wireless networks, many studies have investigated application of RL in the field of massive access management recently [9], [23], [18–21, 33–38]. Different distributed RL frameworks were proposed to address the massive access management problem under massive scale and stringent resource constraints [33], [34], where each device has the ability to intelligently make its informed transmission decision by itself without a central controller. The authors in [9] and [23] adopted the sparse dictionary learning to facilitate massive connectivity for a massive-device multiple access communication system, and the learning structure does not need any prior knowledge of active devices.

Furthermore, the delay-aware access control of massive random access for mMTC and M2M was studied in [33], [35], and [36], and spectrum access algorithms based on RL were proposed to determine the access decision with high successful connections and low network access latency. As future wireless networks are complex and large scale, RL cannot effectively deal with the high-dimensional input state space. DRL (DRL combines deep learning with RL to learn itself from experience) was developed to solve complex spectrum access decision-making tasks under large-state space [18–21, 37, 38].

The authors in [18, 19, 37] proposed distributed dynamic spectrum access (DSA) approaches based on DRL to search the optimal solution for the DSA problem under the large-state space and local observation information. These distributed learning approaches are capable of encouraging devices to make spectrum access decisions according to their own observations without central controller, and hence they have a great potential for finding efficient solutions for real-time services. Hua et al. in [20] presented a network-powered deep distributional Q-network to allocate radio resources for diversified services in 5G networks.

Moreover, Yu et al. in [21] investigated a DRL-based multiple access protocol to learn the optimal spectrum access policy considering service fairness, and Mohammadi et al. in [38] employed a DQN algorithm for cognitive radio underlay DSA which outperforms the distributed multi-resource allocation. However, the above works [18–21, 37, 38] did not investigate how to address the massive access management problem in their presented spectrum access approaches based on DRL, and most of the works did not consider the stringent reliability and latency constraints into the optimization problem.

4.3 System Model and Problem Formulation

We consider a maritime wireless communication network, as shown in Fig. 4.2, which consists of a BS at the center and a large number of ships with each ship being equipped with a single antenna. The ships are mainly divided into two types: cellular ships (C-ship) which communicate with the BS over the orthogonal spectrum subchannels and D2D ships (D-ship) which establish D2D communication links if two of them want to communicate with each other and they are close enough. In the network, D-ships can opportunistically access subchannels of C-ships while ensuring that the generated interference from D2D pairs to C-ships should not affect the QoS requirements of C-ships. We assume that each C-ship can be allocated with multiple subchannels, and each subchannel only serve for at most one C-ship in one time slot. In addition, each D2D pair can share multiple subchannels of C-ships.

Let K, M, and N denote the number of C-ships, D2D pairs, and subchannels, respectively. The sets of the corresponding C-ship , D2D pair, and subchannel are denoted by $\mathscr{K} = \{1, 2, \ldots, K\}$, $\mathscr{M} = \{1, 2, \ldots, M\}$ and $\mathscr{N} = \{1, 2, \ldots, N\}$, respectively. Let Z denote the total number of communication links, $Z = K + M$, and its corresponding communication link set is defined by $\mathscr{Z} = \{1, 2, \ldots, Z\}$. Let h_k and h_m be the channel coefficients of the desired transmission links from the k-th C-ship to the BS and the transmitter to the receiver in the m-th D2D pair, respectively. Denote by $g_{k,m}$, $g_{m,B}$ and $g_{m',m}$ the interference channel gains from

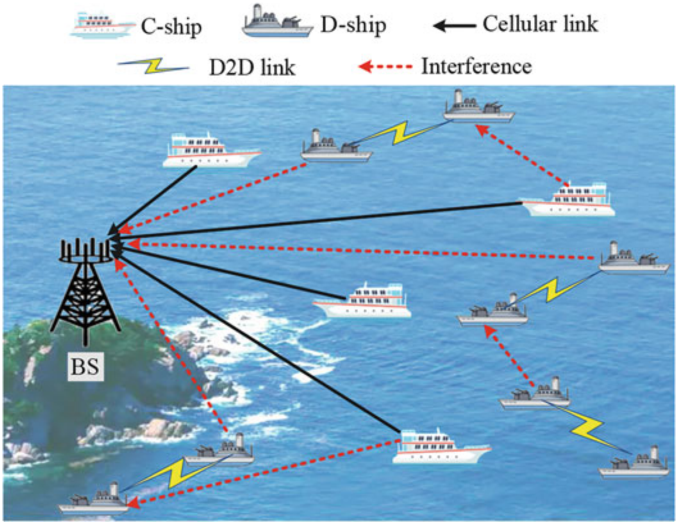


Fig. 4.2 Illustration of maritime wireless communication networks

the k-th C-ship to the receiver of D2D pair m, the transmitter of D2D pair m to the BS, and the transmitter of the m'-th D2D pair to the receiver of the m-th D2D pair, respectively.

In the spectrum reusing case, C-ships suffer co-channel interference from the transmitters of D2D pairs if they share the subchannels with D2D pairs. As a result, the received signal-to-interference-plus-noise ratio (SINR) at the BS for C-ship k on the n-th subchannel is expressed by

$$SINR_{k,n} = \frac{P_{k,n}h_k}{\sum\limits_{m\in\mathscr{M}} \rho_{m,n}P_{m,n}g_{m,B} + \delta_k^2}, \tag{4.1}$$

where $P_{k,n}$ and $P_{m,n}$ denote the transmission power values of the k-th C-ship and the m-th D2D pair's transmitter on the n-th subchannel, respectively. $\rho_{m,n}$ is the subchannel access indicator, $\rho_{m,n} \in \{0, 1\}$; $\rho_{m,n} = 1$ indicates that the m-th D2D pair assigns on the n-th subchannel; otherwise, $\rho_{m,n} = 0$. δ_k^2 is the additive white Gaussian noise power. In (4.1), $\sum_{m\in\mathscr{M}} \rho_{m,n}P_{m,n}g_{m,B}$ is the co-channel interference.

In addition, subchannel sharing also leads to the co-channel interference to D2D pairs, which is the generated interference from the co-channel C-ship and co-channel D2D pairs on the same subchannel. Hence, the received SINR at the m-th D2D pair's receiver when it reuses the n-th subchannel of the k-th C-ship is given by

$$SINR_{m,n} = \frac{P_{m,n}h_m}{P_{k,n}g_{k,m} + \sum\limits_{m' \in \mathscr{M}, m' \neq m} \rho_{m,m',n}P_{m',n}g_{m',m} + \delta_m^2}, \tag{4.2}$$

where $\rho_{m,m',n}$ is the subchannel access indicator, $\rho_{m,m',n} \in \{0, 1\}$; $\rho_{m,m',n} = 1$ indicates that both the m-th D2D pair and m'-th D2D pair assign on the same n-th subchannel in one time slot; otherwise, $\rho_{m,m',n} = 0$. δ_m^2 is the additive white Gaussian noise power.

Then, the data rate of the k-th C-ship and the m-th D2D pair on their assigned subchannels are respectively expressed by

$$R_k = \sum_{n \in \mathscr{N}} \rho_{k,n} \log_2 \left(1 + SINR_{k,n}\right), \tag{4.3}$$

and

$$R_m = \sum_{n \in \mathscr{N}} \rho_{m,n} \log_2 \left(1 + SINR_{m,n}\right), \tag{4.4}$$

where $\rho_{k,n}$ is the subchannel access indicator which has the same definition of $\rho_{m,n}$ and $\rho_{m,m',n}$ as aforementioned above.

4.3.1 Network Requirements

(1) *URLLC Requirements:* In 5G and B5G networks, different ships have different QoS requirements, i.e., some ships have ultrahigh-reliability communication requirements, some ships need strict low-latency services, and even some ships have both the stringent low-latency and high-reliability requirements. For example, intelligent transportation and factory automation have stringent URLLC requirements for real-time safety information exchange or hazard monitoring, where the maximum latency is less than 5 ms (even about 0.1 ms) and the transmission reliability needs to be higher than that $1 - 10^{-5}$ (or even $1 - 10^{-5}$), but they do not need the high data rate.

For URLLC requirements, we assume that the packet arrival process of the i-th ($i \in \mathscr{L}$) communication link is independent and identically distributed and follows Poisson distribution with the arrival rate λ_i [11]. Let L_i denote the packet size in bits of the i-th communication link, and it follows the exponential distribution with mean packet size $\bar{L}_i$. Generally, the total latency mainly includes the transmission delay (T_r), queuing waiting delay (T_w), and processing/computing delay (T_c), which can be expressed by [11]

$$T{=}T_\mathrm{r} + T_\mathrm{w} + T_\mathrm{c}. \tag{4.5}$$

In (4.5), the transmission delay of the packet L_i can be given by $T_{\mathrm{r}} = L_i/(W \times R_i)$, where W is the bandwidth of each subchannel and R_i is the data rate given in (4.3) or (4.4), respectively.

Due to the low-latency constraint, each packet requires to be successfully transmitted in a given time period. Let $T_{\max}$ denote the maximum tolerable latency threshold, so the latency outage probability of URLLC can be given by

$$p_i = \Pr\{T > T_{\max}\} \leq p_{\max}, \tag{4.6}$$

where $p_{\max}$ is the maximum SINR violation probability.

It is hard to directly calculate the ship's packet latency shown in (4.5), and hence the outrage probability in (4.6) is difficult to be achieved. However, we can transform the latency constraint (4.6) into the data rate constraint by using max-plus queuing methods [18].

To guarantee the latency outage probability constraint shown in (4.7), the data rate R_i of each URLLC service of the i-th communication link should meet

$$R_i \geq \frac{\bar{L}_i}{W T_{\max}} \left(F_i - f_{-1}\left(p_{\max} F_i e^{F_i} \right) \right), \tag{4.7}$$

where $f_{-1}(\cdot) : [-e^{-1}, 0) \to [-1, \infty)]$ denotes the lower branch of Lambert function meeting $y = f_{-1}(ye^y)$ [18], $F_i = \lambda_i T_{\max}/(1 - e^{\lambda_i T_{\max}})$, and $R_{i,\min}$ is the minimum data rate to ensure the latency constraint shown in (4.6). The relevant proof of (4.7) can be seen in [18, Th. 2]. If the transmission data rate is less than the minimum data rate threshold, in other words, the latency exceeds the maximum latency threshold, the current URLLC service is unsuccessful and its corresponding packet transmission is stopped.

In addition, the SINR value can be used to characterize the reliability of URLLC. In detail, the received SINR at the receiver should be beyond the minimum SINR threshold. Otherwise, the received signal cannot be successfully demodulated. Hence, the outage probability in terms of SINR can be given by

$$\Pr\{SINR_{i,n} < SINR_{i,n}^{\min}\} \leq p_{\max}, \tag{4.8}$$

where $SINR_{i,n}^{\min}$ denotes the minimum SINR threshold of communication link i on the n-th subchannel and $p_{\max}^{\mathrm{outage}}$ denotes the maximum violation probability.

(2) *Minimum Data Rate Requirements:* In addition to the high-reliability and low-latency requirements mentioned in Sect. 4.2, some C-ships and D2D pairs may have the minimum data rate requirements. Let $R_{k,\min}$ and $R_{m,\min}$ denote the minimum data rate requirements of the k-th C-ship and the m-th D2D pair, respectively. Then, the minimum data rate requirements are given by

$$R_k \geq R_{k,\min}, \ \forall k; \quad R_m \geq R_{m,\min}, \ \forall m. \tag{4.9}$$

4.3.2 Problem Formulation

The objective of this chapter is to maximize the overall network ratio of the sum data rate and the sum energy consumption (EE) while guaranteeing the network requirements. Then, the massive access management problem (joint subchannel access and transmission power control) is formulated as follows:

$$
\begin{aligned}
\max_{\rho, P} \quad & \eta = \frac{\sum\limits_{k \in \mathscr{K}} R_k + \sum\limits_{m \in \mathscr{M}} R_m}{\sum\limits_{n \in \mathscr{N}} \left(\sum\limits_{k \in \mathscr{K}} \rho_{k,n} P_{k,n} + \sum\limits_{m \in \mathscr{M}} \rho_{m,n} P_{m,n} \right) + Z P_c} \\
s.t. \quad & \text{(a)}: \ (4.7),\ (4.8),\ (4.9) \\
& \text{(b)}: \ \rho_{k,n} \in \{0, 1\}, \quad \rho_{m,n} \in \{0, 1\}, \ \forall k,\ m,\ n \\
& \text{(c)}: \ \sum_{k \in \mathscr{K}} \rho_{n,k} \leq 1, \ \forall n \in \mathscr{N} \\
& \text{(d)}: \ \sum_{n \in \mathscr{N}} \rho_{k,n} P_{k,n} \leq P_k^{\max}, \ \forall k \in \mathscr{K} \\
& \text{(e)}: \ \sum_{n \in \mathscr{N}} \rho_{m,n} P_{m,n} \leq P_m^{\max}, \ \forall m \in \mathscr{M},
\end{aligned}
\tag{4.10}
$$

where ρ and P denote the subchannel assignment and power control strategies, respectively. $P_k^{\max}$ and $P_m^{\max}$ denote the maximum transmission power values of each C-ship and each D-ship, respectively. P_c denotes the circuit power consumption of one communication link. Constraint (4.10c) guarantees that each subchannel is allocated at most one C-ship. Constraints (4.10d) and (4.10e) are imposed to ensure the power constraints of ships.

4.4 Problem Transformation

Clearly, the optimization problem given in (4.10) is not easy to be solved as it is a non-convex combination and NP-hard problem. More importantly, the optimization objective is just a single time slot optimization problem, where the massive access decision is only based on the current state with the fixed optimization function. The single time slot massive access decision approaches may converge to the suboptimal solution and obtain the greedy-search like performance due to the lack of the historical network state and the long-term benefit.

Hence, model-free RL as a dynamic programming tool can be applied to address the decision-making problem by learning the optimal solutions over dynamic environment. Similar to most of existing studies, we apply MDP to model the massive access decision-making problem in the RL framework by transforming the optimization problem (4.10) into MDP.

In the MDP model, each communication link acts as an agent by interacting with outside environment, and the MDP model is defined as a tuple $(\mathscr{S}, \mathscr{A}, \mathscr{P}, r, \gamma)$, where $\mathscr{S}$ is the state space set, $\mathscr{A}$ denotes the action space set, $\mathscr{P}$ indicates the transition probability: $\mathscr{P}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ is the probability of transferring from a current state $\mathbf{s}_t \in \mathscr{S}$ to a new state $\mathbf{s}_{t+1} \in \mathscr{S}$ after taking an action $\mathbf{s}_t \in \mathscr{A}$, r denotes the immediate reward, and $\gamma \in (0, 1)$ denotes the discount factor. More details of the MDP model for massive maritime access management are presented as follows.

Agent: Each communication transmitter in the maritime wireless network.

State: In 5G and B5G networks, the network state is defined as $\mathbf{s} = \{\mathbf{s}_{\text{cha}}, \mathbf{s}_{\text{cq}}, \mathbf{s}_{\text{tr}}, \mathbf{s}_{\text{QoS}}\} \in \mathscr{S}$, where $\mathbf{s}_{\text{cha}}$ indicates the subchannel working status (idle or busy), $\mathbf{s}_{\text{cq}}$ is the channel quality (i.e., SINR), $\mathbf{s}_{\text{tr}}$ is the traffic load of each packet, and $\mathbf{s}_{\text{QoS}}$ represents the QoS satisfaction level (the transmission success probability), such as the satisfaction levels of the minimum data rate, latency, and reliability.

Action: For the massive access management problem, each agent will decide which subchannels can be assigned and how much transmit power should be allocated on the assigned subchannels. Hence, the action can be defined as $\mathbf{a} = \{\rho_{\text{cha}}, P_{\text{pow}}\} \in \mathscr{A}$ which includes the subchannel assignment indicator (ρ_{cha}) and the transmission power (P_{pow}). At each time slot, the action of each ship consists of channel assignment indicator $\rho_{\text{cha}} \in \{0, 1\}$ and transmission power level $P_{\text{pow}} \in \{50, 150, 300, 500\}$ in mW where the transmission power is discretized into four levels. The action space of each ship is not large in general, but the overall action space of all the ships in the massive access scenario is large. Hence, we discretize the transmission power levels, e.g., as small as possible, we choose the four transmission power levels instead of more levels.

Reward function: In order to exploit the ship communication experiences, the RL-based communication scheme designs a reward function in the learning process. More specifically, each learning agent searches its decision-making policy by maximizing its reward in the interaction with environment. Hence, it is important to design an efficient reward function to improve the ships' service satisfaction level.

Here, let $\mathscr{Z}'$ denotes the set of communication links in the URLLC scenario where the ships have both the reliability and latency requirements, and $\mathscr{Z}''$ denotes the set of communication links in the normal scenario where the ships have minimum data requirements. $|\mathscr{Z}'| = Z'$ and $|\mathscr{Z}''| = Z''$. Let R_i^{nor} and $R_{i,\min}^{\text{nor}}$ denote the instantaneous data rate and the minimum data rate threshold in the normal scenario, respectively.

According the optimization problem shown in (4.10), considering the different QoS requirements, we design a new QoS-aware reward function for the massive access management problem, where the reward function of the i-th communication link includes the network EE, as well as the reliability, latency, and minimum data rate requirements, which is expressed by

$$r = \eta_{i,EE} - c_1 \chi_i^{\text{URLLC}} - c_2 \chi_i^{\text{nor}}, \tag{4.11}$$

where

$$\chi_i^{\text{URLLC}} = \begin{cases} 0, & \text{if } (4.7) \text{ and } (4.8) \text{ are satisfied,} \\ 1, & \text{otherwise.} \end{cases} \tag{4.12}$$

$$\chi_i^{\text{nor}} = \begin{cases} 1, & \text{if } R_i^{\text{nor}} < R_{i,\min}^{\text{nor}}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.13}$$

In (4.11), the part 1 indicates the immediate utility (network EE), and the part 2 and part 3 are the cost functions of the transmission failures which are defined as the unsatisfied URLLC requirements and the unsatisfied minimum data rate requirements, respectively. The parameters c_i, $i \in \{1, 2\}$ denote the positive constants of the latter two parts in (4.11), and they are adopted for balancing the utility and cost [19], [28], [19].

The objectives of (4.12) and (4.13) are to refract the QoS satisfaction levels of both the URLLC services and normal services, respectively. In detail, if the URLLC requirement of one packet is satisfied in the current time slot, then $\chi_i^{\text{URLLC}} = 0$; if the minimum data rate is satisfied, then $\chi_i^{\text{nor}} = 0$. This means that there is no cost or punishment of the reward due to the successful transmission with QoS guarantees. Otherwise, $\chi_i^{\text{URLLC}} = 1$, or $\chi_i^{\text{nor}} = 1$.

The reward function shown in (4.11) may have the same reward values for some cases. For example, the following two cases may have the same reward for different values: Case I, the URLLC requirement is not satisfied, while the minimum data rate requirement is satisfied, and then $\chi^{\text{URLLC}} = 1$ and $\chi^{\text{nor}} = 0$; Case II, the URLLC requirement is satisfied, while the minimum data rate requirement is not satisfied, and then $\chi^{\text{URLLC}} = 0$ and $\chi^{\text{nor}} = 1$. For these two cases, they may have the same reward function values: $\mathbf{r} = \eta_{EE} - c_1 * 1 - c_2 * 0$ and $\mathbf{r} = \eta_{EE} - c_1 * 0 - c_2 * 1$ with $c_1 = c_2$ being the punishment factors. If the punishment factors $c_1 \neq c_2$, the two cases have different reward function values. We would like to mention that the values of the punishment factors c_1 and c_2 have important impacts on the reward function, if $c_1 > c_2$, the URLLC requirement has the higher impact on the final reward value than that of the minimum data rate requirement; by contract, if $c_1 < c_2$, the minimum data rate requirement has the higher impact on the final reward value than that of the URLLC requirement. Furthermore, if $c_1 = c_2$, both the URLLC requirement and minimum data rate requirement have the same impacts on the reward value.

In RL, each agent in the MPD model tries to select a policy π to maximize a discounted accumulative reward, where π is a mapping from state $\mathbf{s}$ with the probability distribution over actions that the agent can take: $\pi(\mathbf{s}) : \mathscr{S} \to \mathscr{A}$. The discounted accumulative reward is also a called the state-value function for starting the state $\mathbf{s}$ with the current policy π, and it is defined by

$$V^{\pi}(\mathbf{s}) = \left\{ \sum_{t=1}^{\infty} \gamma^t r_t\left(\mathbf{s}_t, \mathbf{a}_t\right) | \mathbf{s}_0 = \mathbf{s}, \pi \right\}. \tag{4.14}$$

The function V^{π} ($\mathbf{s}$) in (4.14) is usually applied to test the quality of the selected policy π when the agent selects the action $mathbfa$. The MPD model tries to search the optimal state-value function $V^*(s)$, which is expressed by

$$V^*(\mathbf{s}) = \max_{\pi} V^{\pi}(\mathbf{s}). \tag{4.15}$$

Once $V^*(\mathbf{s})$ is achieved, the optimal policy $\pi^*\left(\mathbf{s}_t\right)$ under the current state $\mathbf{s}_t$ is determined by

$$\pi^*\left(\mathbf{s}_t\right) = \arg\max_{\mathbf{a}_t \in \mathscr{A}} \bar{U}_t(\mathbf{s}_t, \mathbf{a}_t) + \sum_{\mathbf{s}_{t+1}} P\left(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t\right) V^*\left(\mathbf{s}_{t+1}\right), \tag{4.16}$$

where $\bar{U}_t(\mathbf{s}_t, \mathbf{a}_t)$ denotes the expected reward by selecting action $\mathbf{a}_t$ at state $\mathbf{s}_t$. To calculate $V^*(\mathbf{s})$, the iterative algorithms can be applied. However, it is difficult to get the transition probability $P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ in practical environments, but RL algorithms, such as Q-learning, policy gradient, and DQN, are widely employed to address MDP problems under environment uncertainty.

In Q-learning algorithm, the Q-function is used to calculate the accumulative reward for starting from a state $\mathbf{s}$ by taking an action $\mathbf{a}$ with the selected policy π, which can be given by

$$Q^{\pi}\left(\mathbf{s}, \mathbf{a}\right) = \left\{ \sum_{t=1}^{\infty} \gamma^t r_t\left(\mathbf{s}_t, \mathbf{a}_t\right) | \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a}, \pi \right\}. \tag{4.17}$$

Similarly, the optimal Q-function is obtained by

$$Q^*(\mathbf{s}, \mathbf{a}) = \max_{\pi} V^{\pi}(\mathbf{s}, \mathbf{a}). \tag{4.18}$$

In Q-learning algorithm, the Q-function is updated by

$$Q_{t+1}\left(\mathbf{s}_t, \mathbf{a}_t\right) = Q_t\left(\mathbf{s}_t, \mathbf{a}_t\right) + \alpha\Big(r_{t+1} + \gamma \max_{\mathbf{a}_{t+1}} Q_t\left(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}\right) - Q_t\left(\mathbf{s}_t, \mathbf{a}_t\right)\Big), \tag{4.19}$$

where α denotes the learning rate. When $Q^*(\mathbf{s}, \mathbf{a})$ is achieved, the optimal policy is determined by

$$\pi^*(\mathbf{s}) = \arg\max_{\mathbf{a} \in A} Q^*\left(\mathbf{s}, \mathbf{a}\right). \tag{4.20}$$

4.5 Distributed Cooperative Multi-agent RL-Based Massive Access

Even though Q-learning is widely adopted to design the resource management policy in wireless networks without knowing the transition probability in advance, it has some key limitations for its application in large-scale 5G and B5G networks, such as Q-learning has slow convergence speed under large-state space, and it cannot deal with large continuous state-action spaces. Recently, a great potential is demonstrated by DRL that combines neural networks (NNs) with Q-learning, called DQN, which can efficiently address the abovementioned problems and achieve better performance owing to the following reasons. Firstly, DQN adopts NNs to map from the observed state to action between different layers, instead of using storage memory to store the Q-values. Secondly, large-scale models can be represented from high-dimensional raw data by using NNs. Furthermore, by applying experience replay and generalization capability brought by NNs, DQN can improve network performance.

In 5G and B5G networks shown in Fig. 4.2, massive communication links aim to access the limited radio spectrum, which can be modeled as a multi-agent RL problem, where each communication link is regarded as a learning agent to interact with network environment to learn its experience, and the learned experience is then utilized to optimize its own spectrum access strategy. Massive agents explore the outside network environment and search spectrum access and power control strategies according to the observations of the network state. The proposed deep multi-agent RL-based approach consists of two stages, a training stage and a distributed cooperative implementation stage. The main contributions of the proposed distributed cooperative multi-agent RL-based approach for massive access are provided as follows in detail.

4.5.1 *Training Stage of Multi-agent RL for Massive Access*

For the training stage, we adopt DQN with experience relay to train the multi-agent RL for efficient learning of massive access policies. Figure 4.3 indicates the training process. All communication links are regarded as agents and the wireless network acts as the environment. Firstly, each agent intelligently observes its current state (e.g., subchannel status (busy or idle), channel quality, traffic load, and QoS satisfaction levels) by integrating with the environment. Then, it makes decision and chooses one action according to its learned policy. After that, the environment feedbacks a new state and an immediate reward to each agent. Based on the feedback, all agents smartly learn new policies in the next time step. The optimal parameters of DQN can be trained with an infinite number of time steps. In addition, the experience replay mechanism is adopted to improve the learning speed, the learning efficiency, and the learning stability toward the optimal policy for the

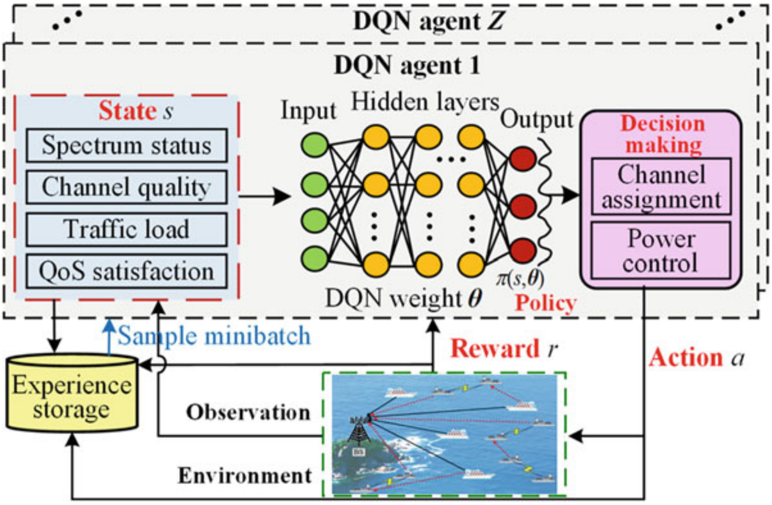


Fig. 4.3 DQN training-based intelligent subchannel assignment and power control for massive access

massive access management. The training data is stored in the storage memory, and a random mini-batch data is sampled from the storage memory and used to optimize DQN.

At each training or learning step, each DQN agent updates its weight, θ , to minimize the loss function defined by

$$Loss(\theta_t) = \left(r_{t+1}(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{a\in\mathscr{A}} Q_t\left(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \theta_t\right) - Q_t\left(\mathbf{s}_t, \mathbf{a}_t, \theta_t\right)\right)^2. \tag{4.21}$$

One important reason of adopting DQN is to update the loss functions given in (4.21) at each tainting step to decrease the computational complexity for large-scale learning problems [19–23]. The DQN weight θ is obtained by using the gradient descent method, which can be expressed as

$$\theta_{t+1} = \theta_t + \beta\nabla Loss(\theta_t), \tag{4.22}$$

where β denotes the learning rate of the weight θ and $\nabla(.)$ is the first-order partial derivative.

Then, each agent selects its action according to the selected policy $\pi(s_t, \theta_t)$, which is given by

$$\pi(\mathbf{s}_t, \theta_t) = \arg\max_{\mathbf{a}\in\mathscr{A}} Q_t\left(\mathbf{s}_t, \mathbf{a}_t, \theta_t\right). \tag{4.23}$$

Algorithm 3 DQN Training Stage of Subchannel Assignment and Power Control with Multi-agent RL for Massive Access

```
Input: DQN structure, environment simulator and QoS requirements of all ships (e.g., reliability, latency and minimum data rate).
for each episode $j$=1,2,..., $N^{\text{epi}}$ do
  Initialize: Initial Q-networks for all agents (e.g., Q-function $Q(\mathbf{s}, \mathbf{a})$, policy strategy $\pi(\mathbf{s}, \mathbf{a})$, and weight $\theta$) and experience replay $D$.
  for each iteration step $t$=0,1,2,..., $T$ do
    Each agent observes its state $\mathbf{s}_t$;
    Select a random action $\mathbf{a}_t$ with the probability $\varepsilon$;
    Otherwise, choose the action $\mathbf{a}_t = \arg\max_{\mathbf{a}\in\mathscr{A}} Q_t(\mathbf{s}_t, \mathbf{a}_t, \theta_t)$;
    Execute action $\mathbf{a}_t$, then obtain a reward $\mathbf{r}_t$ by (4.15), and observe a new state $\mathbf{s}_{t+1}$;
    Save experience $e_t = (\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})$ into the storage memory $D$;
  end for
  for each agent do
    Sample a random mini-batch data $e_t$ from $D$;
    Update the loss function by (4.21);
    Perform a gradient descent step to update $\theta_{t+1}$ by (4.22);
    Update the policy $\pi$ with maximum Q-value by (4.23), and chose an action based on $\pi$;
  end for
end for
return: Return trained DQN models.
```

Pseudocode for training DQN is presented in Algorithm 3. The communication environment contains both the C-ships and D-ships and their positions in the served coverage area of the BS, and the channel gains are generated based on their positions. Each agent has its trained DQN model that takes as input of current observed state s_t and outputs the Q-function with the selected action a_t. The training loop has a finite number of episodes N^{epi} (i.e., tasks) and each episode has T training iterations. At each training step, after observing the current state s_t, all agents explore the state-action space by applying the $\varepsilon-$ greedy method, where each action a_t is randomly selected with the probability ε_t, while the action is chosen with the largest Q-value $Q_t(s_t, a_t, \theta_t)$ with the probability $1 - \varepsilon_t$. After executing a_t (subchannel assignment and power control), agents will receive an immediate reward r_t and observe a new state s_{t+1} from the environment. Then, the experience $e_t = (s_t, a_t, r(s_t, a_t), s_{t+1})$ is stored into the replay memory D. At each episode, a mini-batch data from the memory is sampled to update the weight θ_t of DQN.

4.5.2 Distributed Cooperative Implementation of Multi-agent RL for Massive Access

The abovementioned trained DQN models with the computation intensive training procedure are shown in Sect. 4.4, which can be completed offline at BS since BS has powerful computing capacity to train large-scale models. After adequate

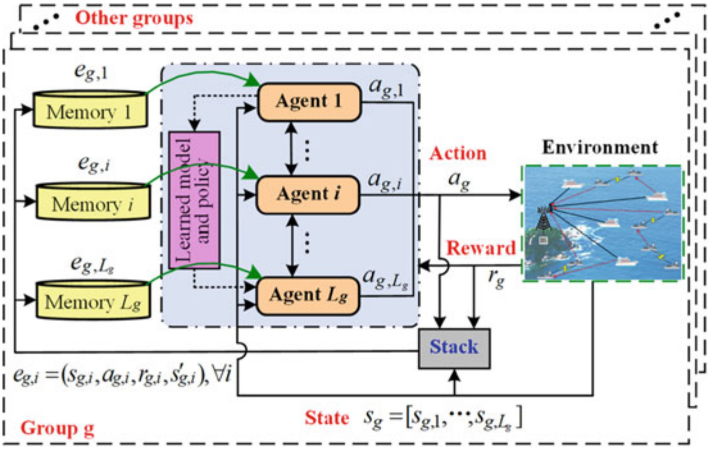


Fig. 4.4 Distributed cooperative multi-agent RL framework

training, the trained models are utilized for implementation. Here, we propose a distributed cooperative learning approach to optimize the network performance in massive access scenario.

During the distributed cooperative implementation stage, at each learning step, each communication link (agent) utilizes its local observation and information to choose its action with the maximum Q-value. In this case, each agent has no knowledge of actions chosen by other agents if the actions are updated simultaneously, and new joined agents need to train their own learning model with extra training computational time or cost. In order to address this issue, motivated by the concept of transfer learning and cooperative learning, we present a distributed cooperative learning approach to improve the learning efficiency and enhance the service performance of each agent, where ships are encouraged to communicate and share their learned experiences and decisions within a small number of neighbors, and finally learn with each other, as shown in Figs. 4.3 and 4.4.

(1) *Transfer Learning:*

(i) **The Expert Agent Selection:** When a new ship joins 5G and B5G networks, or one ship applies a new communication service, instead of building a new learning model, it can communicate with neighboring ships to search one suitable expert to utilize the expert's current learning model. In addition, if one communication link has poor performance (e.g., low convergence speed and poor QoS satisfaction levels) according to its current learning strategy, it can search one neighboring communication link (agent) as the expert and then utilizes the learned model or policy from the expert.

Generally, to find the expert, ships exchange the following several metrics with their neighbors: (a) the types of ship, e.g., C-ship and D2D ship; (b) the communication services, which mainly refer to URLLC service and normal service; and (c) the related QoS parameters, such as the target thresholds of reliability and latency, and the minimum data rate. The similarity of the agents can be evaluated by adopting the manifold learning, which is also called the Bregman ball [19]. The Bregman ball is defined as the minimum manifold with a central Θ_{cen} (the information of the learning agent, where information refers to the types of ship, communication services, and QoS parameters mentioned above), and a radius Ψ_{rad}. Any information point Θ_{poi} (the information of neighbors) is inside this ball, and the agent tries to search the information point which has the highest similarity with Θ_{cen}. The distance between any point and the central Θ_{cen} is defined by

$$\text{Dis}\left(\Theta_{\text{cen}}, \Psi_{\text{rad}}\right) = \left\{\Theta_{\text{poi}} \in \Theta : \text{Dis}\left(\Theta_{\text{poi}}, \Theta_{\text{cen}}\right) \leq \Psi_{\text{rad}}\right\}. \tag{4.24}$$

After the highest similarity level (the smallest distance achieved by (4.24)) between the learning agent and the expert agent is found, the learning agent can use the learned DQN model of the selected expert agent.

(ii) **Learning from Expert Agent:** As analyzed above, after finding the expert agent, the learning agent uses the transferred DQN model $Q^{\mathrm{T}}(s, a)$ from the expert agent and its current native DQN model $Q^{\mathrm{C}}(s, a)$ to generate an overall DQN model. Accordingly, the new Q-table of the learning agent can be expressed as

$$Q^{\mathrm{N}}(\mathbf{s}, \mathbf{a}) = \mu Q^{\mathrm{T}}\left(\mathbf{s}, \mathbf{a}\right) + \left(1 - \mu\right) Q^{\mathrm{C}}\left(\mathbf{s}, \mathbf{a}\right), \tag{4.25}$$

where $\mu \in [0, 1]$ is the transfer rate, and it will be gradually decreased after each learning step to reduce the effect of the transferred DQN model from the expert agent on the new DQN model.

In the distributed cooperative manner, the policy vector of all agents are updated as follows:

$$\pi_{t+1}\left(s_t\right) = \begin{bmatrix} \pi_{t+1}^{1}\left(s_t^{1}\right) \\ \vdots \\ \pi_{t+1}^{i}\left(s_t^{i}\right) \\ \vdots \\ \pi_{t+1}^{Z}\left(s_t^{Z}\right) \end{bmatrix} = \begin{bmatrix} \arg\max\limits_{a^{1} \in \mathscr{A}^{1}} Q_{t+1}^{1}\left(\mathbf{s}_t^{1}, \mathbf{a}_t^{1}\right) \\ \vdots \\ \arg\max\limits_{a^{i} \in \mathscr{A}^{i}} Q_{t+1}^{i}\left(\mathbf{s}_t^{i}, \mathbf{a}_t^{i}\right) \\ \vdots \\ \arg\max\limits_{a^{Z} \in \mathscr{A}^{Z}} Q_{t+1}^{Z}\left(\mathbf{s}_t^{Z}, \mathbf{a}_t^{Z}\right) \end{bmatrix}, \tag{4.26}$$

where $Q_{t+1}^{i}(\mathbf{s}_t^{i}, \mathbf{a}_t^{i}, \theta_t^{i})$ denotes the Q-function of the i-th agent (communication link) with its current state-action pair $(\mathbf{s}_t^{i}, \mathbf{a}_t^{i})$ at the current time slot in its DQN model.

When the state-action pairs are visited for many enough times for convergence, all Q-tables will converge to the final point Q^*. Hence, we can get the final learned policy as follows:

$$\pi^*(\mathbf{s}) = \begin{bmatrix} \arg\max_{\mathbf{a}^1 \in \mathscr{A}^1} Q^{1*}\left(\mathbf{s}^1, \mathbf{a}^1\right) \\ \vdots \\ \arg\max_{\mathbf{a}^Z \in \mathscr{A}^Z} Q^{Z*}\left(\mathbf{s}^Z, \mathbf{a}^Z\right) \end{bmatrix}. \tag{4.27}$$

(2) *Cooperative Learning*

If the action is chosen independently according to the local information, each communication link has no information of actions selected by other communication links when the actions are updated simultaneously. Consequently, the states observed by each communication link may fail to fully characterize the environment. Hence, cooperation and decision sharing among agents in the proposed distributed learning approach can improve the network performance, where a small number of communication links will share their actions with their neighbors. In the cooperative manner, the massive number of agents can be classified into G groups, where the g-th group consists of L_g agents and the agents in the same group are also their neighboring agents. The group division principle can adopt the studies in [13].

In general, it is possible to approximate the sum utility of the g-th group $Q_g(\mathbf{s}_g, \mathbf{a}_g)$ by the sum of each agent's utility $Q_{g,i}(\mathbf{s}_{g,i}, \mathbf{a}_{g,i})$ in the same group, where $\mathbf{s}_g$ and $\mathbf{a}_g$ denote the entire state and action of the g-th group, respectively; $\mathbf{s}_{g,i}$ and $\mathbf{a}_{g,i}$ are the individual state and action of the i-th agent in the g-th group, respectively. Hence, the total utility in a small group g can be calculated by

$$Q_g\left(\mathbf{s}_g, \mathbf{a}_g\right) = \sum_{i=1}^{L_g} \left(Q_{g,i}\left(\mathbf{s}_{g,i}, \mathbf{a}_{g,i}\right)\right). \tag{4.28}$$

Then, the joint optimal policy learned in the g-th group can be expressed by

$$\pi_g\left(\mathbf{s}_g\right) = \arg\max_{\mathbf{a}_g \in \mathscr{A}_g} \left(Q_g\left(\mathbf{s}_g, \mathbf{a}_g\right)\right), \tag{4.29}$$

where $\mathscr{A}_g$ denotes the entire action space of the g-th group.

In fact, the cooperation can be defined by allowing communication links (agents) to share their selected actions with their neighboring links and take turns to make decisions, which can enhance the overall feedback reward by choosing the actions jointly instead of independently. For example, in the fully distributed learning manner, each spectrum access may run into collisions when other links make their decisions independently and happen to assign the same subchannel, leading to the increased co-channel interference and reduce the performance. By contrast, in the

Algorithm 4 Distributed Cooperative Implementation of Multi-agent RL for Massive Access

```
1: Input: DQN structure, environment simulator and QoS requirements of all ships.
2: start: Load DQN models.
3: loop
4:   Each agent (communication link) observes its state $\mathbf{s}$;
   Transfer learning
5:   if the agent is new, or needs new service or has poor performance, then;
6:     The agent exchanges information with its neighbors;
7:     Search the expert with the highest similarity by (4.24);
8:     Use the learned model from the expert;
9:     Update the overall Q-table by (4.25);
10:    Update the transfer rate $\mu$, and select an action by (4.31);
11:    Perform learning from step 13 to step 16;
12:  else
   Cooperative learning
13:    In each group $g$, each agent shares its observations and actions;
14:    Each group calculate its cooperative Q-table by (4.28);
15:    Update the joint policy $\pi_g(\mathbf{s}_g)$ with the largest cooperative Q-value $Q_g(\mathbf{s}_g, \mathbf{a}_g)$, and select the joint action $\mathbf{a}_g$;
16:    Execute action $\mathbf{a}_g$, then obtain a reward $r'_g$ using (4.11), and observe a new state $\mathbf{s}'_g$;
17:  end if
18:  Both transfer learning and cooperative learning are jointly updated to optimize the learned policy;
19: end loop
20: output: Subchannel assignment and power control.
```

cooperative learning scenario, to avoid such situation, each communication link has information of the neighbors' actions in its observation and tries to avoid the assignment of the same subchannel in order to achieve more rewards.

The distributed cooperative implementation of multi-agent RL for massive access is shown in Algorithm 4. Generally, at each time step, after observing the states (subchannel occupation status, channel quality, traffic load, QoS satisfaction level, etc.) from the environment, the actions (massive subchannel assignment and power control) in communication links are selected with the maximum Q-value given by loading the trained DQN models shown in Algorithm 3. As mentioned above, a small number of neighboring ships are encouraged to cooperate with each other in the same group to maximize the sum Q-value shown in (4.28), where their decisions are shared in the same group and the joint action strategy a_g is selected with the maximum cooperative Q-value. In addition, it is worth noting that if a new ship joins the network or applies a new service, or one communication link achieves poor performance (e.g., low transmission success probability or low convergence speed), then it can directly search the expert agent from the neighbors in the same group and utilizes the transfer learning model and policy from the expert agent. Finally, all communication links begin transmission with the subchannel assignment and transmission power strategies determined by their learned policies.

Independent DQN is one of the multi-agent reinforcement learning techniques, where each agent independently learns its own policy and considers other agents as part of the environment. Moreover, the combination of experience replay with independent DQN appears to be problematic: the non-stationarity introduced by independent DQN. Hence, we have presented a distributed cooperative multi-agent DQN scheme, and ships are encouraged to communicate and share their learned experiences and actions within a small number of neighbors and finally learn with each other. In this case, the scheme is capable of avoiding the non-stationarity of independent Q-learning by having each agent learn a policy that conditions on an information sharing of the other agents' policies (behaviors) in the same group.

4.5.3 Computational Complexity Analysis

For the training phase, in trained DQN models, let L, B_0, and B_l denote the training layers which are proportional to the number of states, the size of the input layer, and the number of neurons used in DQN, respectively. The complexity in each time step for each agent is calculated by $O\left(B_0B_1+\sum_{l=1}^{L-1}B_lB_{l+1}\right)$ at each training step. In the training phase, each mini-batch has episodes N^{epi} with each episode being T time steps, and each trained model is completed over I iterations until convergence and the network has Z agents with the Z trained DQN models. Hence, the total computational complexity is $O\left(ZIN^{\text{epi}}T(B_0B_1+\sum_{l=1}^{L-1}B_lB_{l+1})\right)$. The high computational complexity of the DQN training phase can be performed offline for a finite number of episodes at a powerful unit (such as the BS) [20], [21].

For the distributed cooperative phase (also called testing phase), our proposed approach applies the transfer learning mechanism and allows the expert agent to share the learned knowledge or actions with other agents. Let $\mathscr{S}'$ and $\mathscr{A}'$ denote the stored state space and action space, respectively. The computational complexities of the classical DQN approach (the fully distributed DQN approach) and the proposed approach are $O(|S|^2\times|\mathscr{A}|)$ and $O(|\mathscr{S}'|^2\times|\mathscr{A}'|+|\mathscr{S}|^2\times|\mathscr{A}|)$ [19], respectively, indicating that the complexity of the proposed approach is higher than the classical DQN learning approach. Nevertheless, the stored state space and action space in the memory is not large at each ship, and hence the complexity of the proposed learning approach is slightly higher than the classical DQN approach. For cooperative learning, a small number of agents in each same group will select their actions jointly instead of independently by sharing their own selected action. Let $\mathbf{a}_{g,i}^{co}$ denote the shared action set of each i-th agent in the g-th group in the current time slot, and then the computational complexity of the g-th group in terms of action sharing is $O\left(\sum\limits_{i=1}^{L_g}|\mathbf{a}_{g,i}^{co}|\right)$. As the network has G groups, the total computational complexity of the cooperative learning is $O\left(\sum\limits_{g=1}^{G}\sum\limits_{i=1}^{L_g}|\mathbf{a}_{g,i}^{co}|\right)$.

4.6 Simulation Results and Analysis

Here, simulation results are provided to evaluate the proposed distributed cooperative multi-agent RL-based massive access approach in a maritime communication network. We consider a single cell with a cell radius of 500 m, and the total number of ships is 150. In addition, we set one fifth of the total number of ships to be normal services and the minimum data rate requirement is set as 3.5 bps/Hz. The maximum D2D communication distance is 75 m. The carrier frequency is 2 GHz, and the total bandwidth is 10 MHz which is equally divided into 20 subchannels with each subchannel having 0.5 MHz. For the URLLC services, the SINR threshold is 5 dB, the processing/computing delay T_{pc}= 0.3 ms, the reliability requirement varies between 99.9% and 99.99999%, and the maximum latency threshold varies between 1 and 10 ms for different simulation settings. The maximum transmit power of each ship and circuit power consumption are 500 mW and 50 mW, respectively. The background noise power is −114 dBm. Each packet size in URLLC links is 1024 bytes. The DQN model consists of three connected hidden layers, containing 250, 250, and 100 neurons, respectively. The learning rate is $\alpha = 0.02$ and the discount factor is set to be $\gamma = 0.95$. The simulation parameters are shown in Table 4.2.

We compare the proposed distributed cooperative multi-agent RL-based massive access approach (denoted as proposed DC-DRL MA, which adopts both transfer learning and cooperative learning mechanisms) with the following approaches:

(1) The group-based massive access approach, where ships are grouped by the similarities with each group having one group leader to communicate with the centralized controller. Then, the subchannel assignment and transmission power control are adjusted iteratively to the communication links in each group,

Table 4.2 Simulation parameters

Parameters	Value
Cell radius	500 m
Carrier frequency	2 GHz
Bandwidth	100 MHz
Maximum D2D communication distance	75 m
Total number of devices	150
Number of subchannels	20
Reliability requirement in URLLC links	99.9%, 99.99%,. . . , 99.99999%
Latency threshold	1, 2, 4, 6, 8, 10 ms
Minimum capacity of each normal link	3.5 bps/Hz
SINR threshold in URLLC links	5 dB
Maximum transmit power of each device	500 mW
The circuit power consumption of each device	50 mW
Background noise power	−114 dBm
Each packet size in URLLC links	1024 bytes

similar to the group-based preamble reservation access approach (denoted as centralized G-MA).

(2) The fully distributed multi-agent RL-based massive access approach (denoted as fully D-DRL MA), similar to the approach, where each communication link selects its subchannel assignment and transmission power strategy based on its own local information without cooperating with other communication links.

(3) Random massive access approach (denoted as random MA), where each communication link chooses its subchannel assignment and transmission power strategy in a random manner.

4.6.1 Convergence Comparisons

Here, we show in Fig. 4.5 the EE with increasing training episodes to investigate the convergence behavior of the proposed multi-agent DQN approach and compared approaches. Clearly, the proposed learning approach significantly achieves the higher EE performance than that of the fully distributed DRL approach [37] and random MA approach. Especially, the proposed approach has faster convergence speed and less fluctuations by adopting transfer learning and cooperative learning mechanisms to improve the learning efficiency and convergence speed. The fully distributed DRL approach [37] is simple without any cooperation among ships, but it achieves poor global performance, leading to the poor EE value. Even though the random MA approach has the simplest structure, the worst performance fails to optimize the network energy efficiency with increasing training episodes. Our proposed approach applies both the transfer learning and cooperative learning mechanisms to enhance the convergence speed and learning efficiency, and the optimized strategy can be learned after a number of training episodes. From Fig. 4.5,

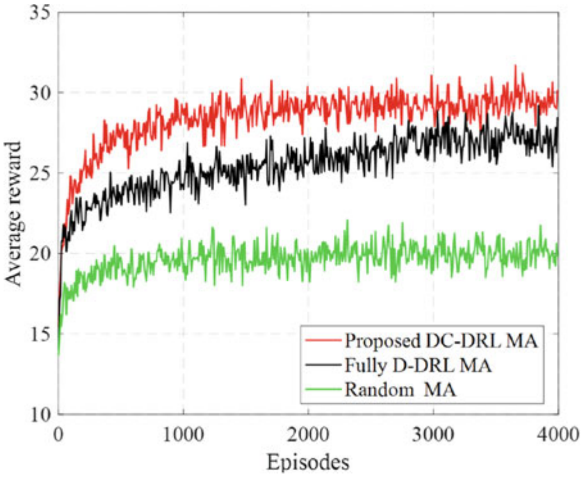


Fig. 4.5 Convergence comparisons of compared learning approaches

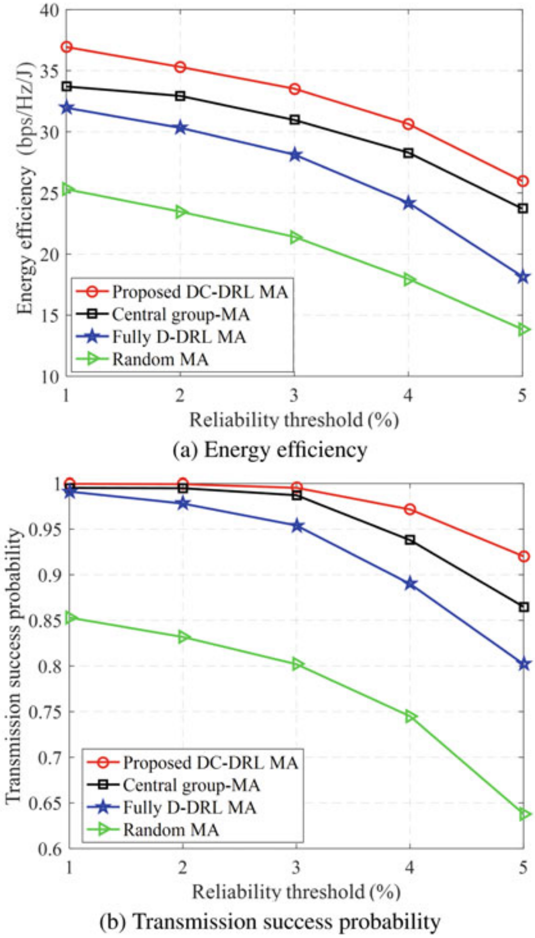
the energy efficiency per episode improves as training continues, demonstrating the effectiveness of the proposed training approach. When the training episode approximately reaches 1900, the performance gradually converges despite some fluctuations due to mobility-induced channel fading in mobile environments. Since we investigate the resource management in massive access scenario, the environment is complex, and the action and state spaces are large for all mobile ships, so our presented learning approach requires about 2000 training episodes to appropriately converge.

4.6.2 Performance Comparisons Under Different Thresholds of Reliability and Latency

Figures 4.6 and 4.7 compare the performances of all approaches under different values of the reliability and latency thresholds, respectively, when the packet arrival rate is 0.03 packets/slot/per link and the total number of ships is 2000. From both Figs. 4.6 and 4.7, for all approaches, we can find that both the EE performance and the transmission success probability drop as the required reliability value increases and the maximum latency threshold decreases. The reason is that the more stringent the reliability and latency constraints are, the worse network EE and transmission success probability the network can archive. In this case, both the transmission power and subchannel assignment strategy need to be carefully designed to guarantee the stringent reliability and latency constraints, such that the transmission success probability can be guaranteed at a high level.

We also observe from Figs. 4.6b and 4.7b that within a reasonable region of the reliability and latency threshold change, the three approaches (except the random search approach) can till achieve the high transmission success probability, which, however, have more unsatisfied transmission link events happen if the constraints are extremely stick (e.g., the reliability threshold grows beyond 99.999% or the maximum latency threshold is less than 4 ms). Compared with other approaches, our proposed approach achieves the higher EE performance and transmission success probability under different reliability and latency requirements, especially the performance gap between the proposed approach and other approaches becomes more significant when the constraints become more stringent. The reason is that our proposed approach employs both the transfer learning and cooperative learning mechanisms to optimize the global subchannel assignment and transmission power strategy, thereby improving the network performance. From Figs. 4.6a and 4.7a, an interesting observation is that compared with the centralized G-MA approach and random MA approach, the EE value curve declines more quickly in our proposed approach when the constraints become stricter. The reason is that the proposed approach designs the specific QoS-aware reward function shown in (4.15) to try to guarantee QoS requirements (meeting the high transmission success probability),

Fig. 4.6 Performance comparisons vs. different reliability thresholds



(a) Energy efficiency

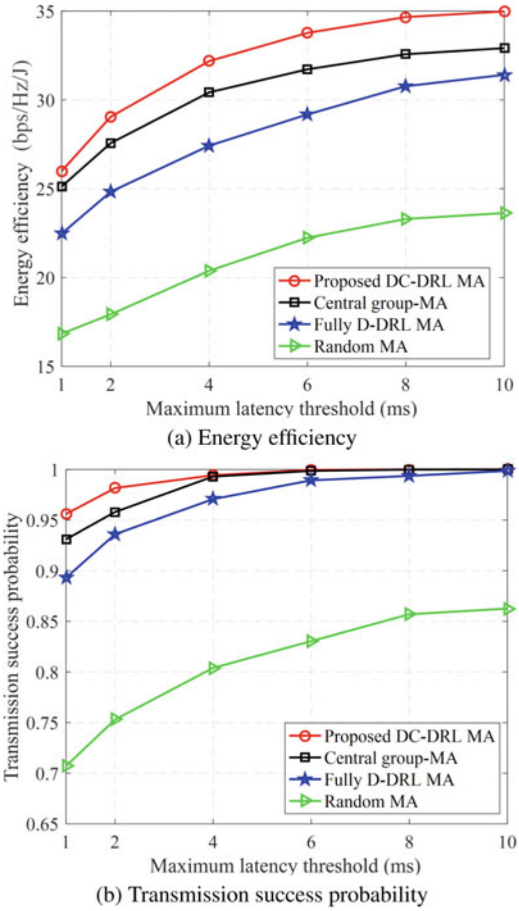
(b) Transmission success probability

and hence the network may sacrifice the part of EE performance to support more successful transmission communication links.

4.7 Intelligent Transmission Scheduling in Maritime Communications

In this chapter, we also consider an actor-critic deep reinforcement learning approach, called AC-DRL in [39] for the intelligent transmission scheduling in maritime communication networks. In the maritime communication networks as shown in Fig. 4.8, all smart devices as learning agents interact with the network as the learning environment.

Fig. 4.7 Performance comparisons vs. different latency thresholds



(a) Energy efficiency

(b) Transmission success probability

Each device cognitively observes its current network state, e.g., channel status (busy or idle), channel quality, devices priority, and traffic load, and chooses an action based on the learned policy strategy independently. Upon performing the communication action, the device observes the new network state and receives an immediate reward from the environment including the previous communication performance sent by the receiver from the feedback in each time slot.

The parameters of both the actor and the critic network in the deep reinforcement learning-based communication scheme as illustrated in Fig. 4.8 can be optimized after a number of learning steps, and the AC-DRL-based communication scheme converges to the optimum value function and policy corresponding to the optimal scheduling policy in the maritime wireless networks. Compared with existing wireless networks, the advantage of the AC-DRL-based approach enables each device to optimize its scheduling policy independently based on the local observation

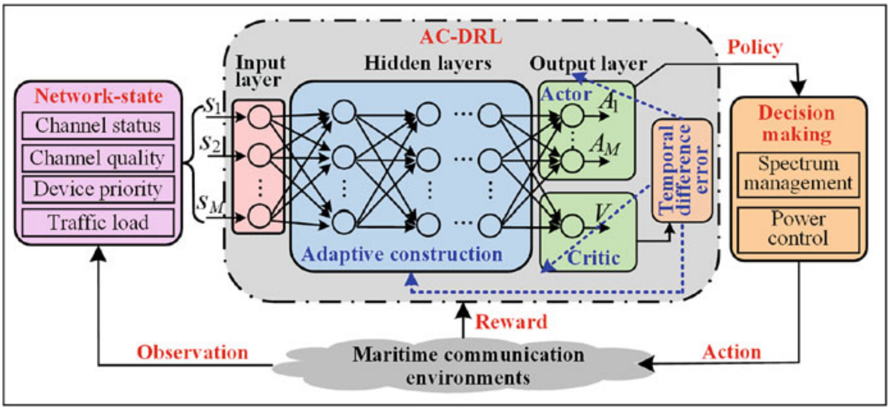


Fig. 4.8 AC-DRL learning framework for intelligent transmission scheduling

information and saves the bandwidth and energy consumption from the continuous exchange of the local information among the neighboring wireless devices.

The AC-DRL framework inputs the network state vector with the number of M network states and outputs the estimated functions of both the actor function vectors A and the critic function V. The hidden layers perform computations on the weighted inputs (i. e., the network states) and produce the net input, which is then applied with activation function to produce the actual output, which is the function of both the actor and the critic network.

In maritime wireless networks, the optimal policy in the spectrum access, data rate control, spectrum allocation, and power control can be obtained via trial and error to support the quality of service requirements. Generally, the learning mechanism is driven by the reward, which can be chosen as a weighted sum of the performance such as the successful packet transmission rate, the energy consumption, and the outage probability in wireless networks [39]. The policy is randomly chosen from the action set at the beginning and then optimized over time slots.

We evaluate the performance of the AC-DRL approach and compare it with the classical AC approach based on PG, denoted as AC-PG, DQN, and random search. In the simulations, the maritime communication devices are assumed to be randomly distributed in a circular cell area with a radius of 500 m, and each aims to send packets with the packet number following a Poisson process. The devices are divided into two groups with different priority levels, and the resource blocks (RBs) are first allocated to the devices with higher priority. The main simulation parameters are listed in Table 4.3 similar to [40].

Figure 4.9 shows the learning process of the four maritime communication scheduling approaches in terms of the reward performance for 600 maritime devices. Three RL approaches significantly outperform the random search, and the AC-DRL approach achieves the best performance with the fastest convergence rate. The

Table 4.3 Simulation parameters

Parameters	Values
Number of devices	200,400,...,1200
Channel model	Frequency selective fading
Packet size	2000 bits
Buffer size	10 packets
Time slot duration	10 ms
Packet arrival rate	0.01/(10 ms)
Power consumption in “active” status	35 mW
Power consumption in “sleep” status	1 mW
Background noise power	−114 dBm
Number of time slots	2000
Number of RBs	16
Number of hidden layers	3
Learning rate of NN	0.02
Training error accuracy	1×10^{-4}
Discount factor	0.002

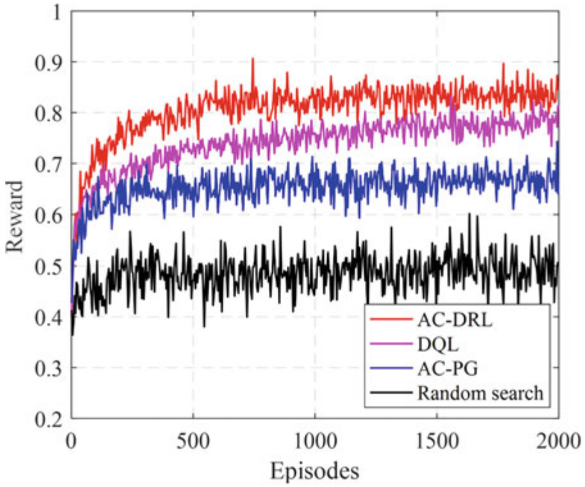


Fig. 4.9 Reward of the maritime communication scheduling approaches over time

DQN approach searches the Q-function approximator and thus sometimes fails to optimize the communication policy under massive devices. In addition, the AC-PG approach has fast convergent rates, but sometimes converges to the local optimal point.

In Figs. 4.10, 4.11, 4.12, and 4.13, the performance of the four approaches with the range of the number of devices in maritime communication networks. The network resource is limited and fixed, when a large number of device packets need to be transmitted as the increase of devices, which results in the frequent handover process, blocking, and retransmission. All these factors increase the average packet

Fig. 4.10 Packet transmission latency of the four maritime communication scheduling schemes

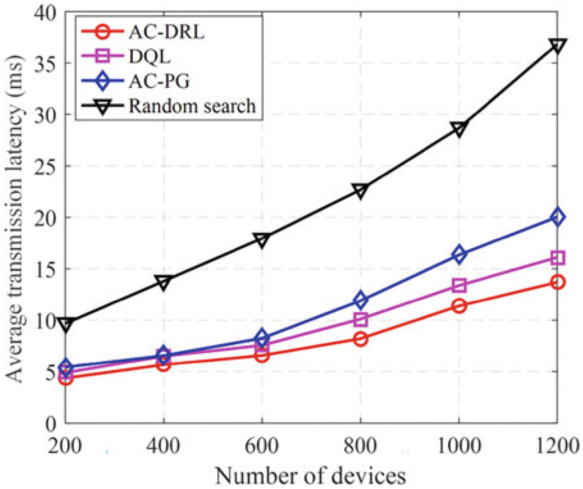
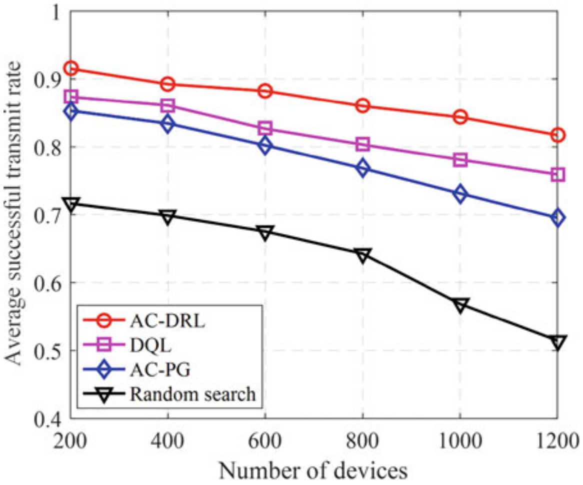


Fig. 4.11 Successful transmit rate



latency and decrease the successful transmission probability; thus, lower reward is obtained in the large number of device regions. In addition, the high frequent handover and retransmission increase the extra power consumption. When the number of devices is 1200, the average transmission latency of AC-DRL approach is 61.6% lower than random search and power consumption of AC-DRL approach is 38.8% lower. With the increase of the number of devices from 200 to 1200, the successful transmit rate of the AC-DRL approach decreases by 10.4% and normalized reward of the AC-DRL approach decreases by 10.1%, which is 48.0% and 48.1% higher than the random search approach, respectively.

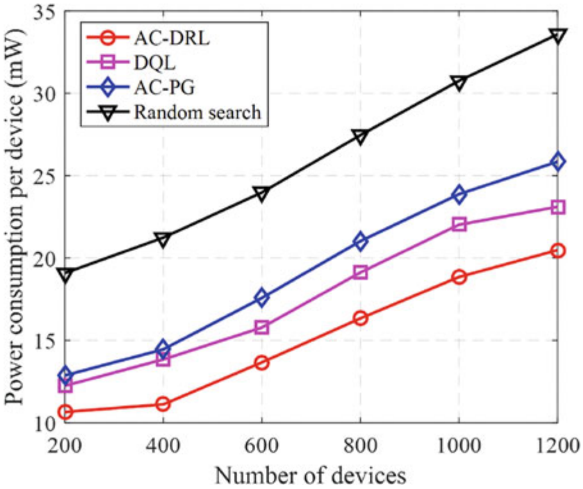


Fig. 4.12 Performance comparison of the average power consumption per device

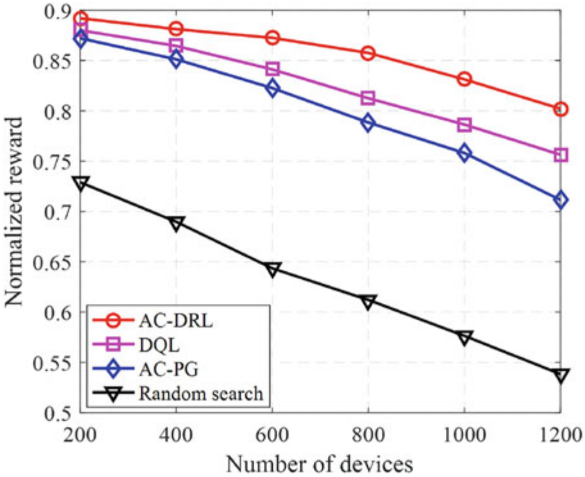


Fig. 4.13 Performance comparison of the normalized reward

4.8 Conclusion

In this chapter, a distributed cooperative channel assignment and power control approach based on multi-agent RL has been presented to solve the spectrum access management problem in maritime wireless communications, where the proposed approach is capable of supporting different QoS requirements (e.g., URLLC and minimum data rate) of a huge number of ships. The proposed multi-agent RL-based approach consists of a centralized training procedure and a distributed cooperative implementation procedure. In order to improve the network performance and QoS satisfaction levels, the transfer learning and cooperative learning mechanisms have been employed to enable communication links to work cooperatively in a distributed cooperative way. Simulation results have confirmed the effectiveness of the proposed learning approach and also showed that the proposed approach outperforms other existing approaches in maritime wireless communication scenarios. Furthermore, an exemplary case study and simulation analysis on the intelligent transmission scheduling are provided to demonstrate the advantage and significance of machine learning in intelligent maritime wireless communications. In a nutshell, machine learning-based physical layer design, decision-making, network management, and resource optimization are exciting areas for future intelligent maritime communications.

References

1. M. R. Palattella, et al., Internet of Things in the 5G era: enablers, architecture, and business models. IEEE J. Sel. Areas Commun. **34**(3), 510–527 (2016)
2. S. Jo, W. Shim, LTE-maritime: high-speed maritime wireless communication based on LTE technology. IEEE Access **7**, 53,172–53,181 (2019)
3. L. Liu, E.G. Larsson, W. Yu, P. Popovski, C. Stefanovic, E. de Carvalho, Sparse signal processing for grant-free massive connectivity: a future paradigm for random access protocols in the Internet of Things. IEEE Signal Procss. Mag. **35**(5), 88–99 (2018)
4. P. Popovski, Ultra-reliable communication in 5G wireless systems, in *Proceeding International Conference 5G Ubiquitous Connectivity* (2014)
5. Y. Kim, Y. Song, S.H. Lim, Hierarchical maritime radio networks for internet of maritime things. IEEE Access, **7**, 54,218–54,227 (2019)
6. T. Yang, H. Liang, N. Cheng, R. Deng, X. Shen, Efficient scheduling for video transmissions in maritime wireless communication networks. IEEE Trans. Veh. Technol. **64**(9), 4215–4229 (2015)
7. T. Yang, Z. Zheng, H. Liang, R. Deng, N. Cheng, X. Shen, Green energy and content-aware data transmissions in maritime wireless communication networks. IEEE Trans. Intelligent Transp. Syst. **16**(2), 751–762 (2015)
8. X. Huang, K. Wu, M. Jiang, L. Huang, J. Xu, Distributed resource allocation for general energy efficiency maximization in offshore maritime device-to-device communication. IEEE Wireless Commun. Lett. **10**(6), 1344–1348 (2021)
9. L. Bai, R. Han, J. Liu, J. Choi, W. Zhang, Random access and detection performance of Internet of Things for smart ocean. IEEE Internet Things J. **7**(10), 9858–9869 (2020)
10. T. Taleb, A. Kunz, Machine type communications in 3GPP networks: potential, challenges, and solutions. IEEE Commun. Mag. **50**(3), 178–184 (2012)
11. M. Lee, Y. Kim, Y. Piao, T. Lee, Recycling random access opportunities with secondary access class barring. IEEE Trans. Mobile Comput. **19**(9), 2189–2201 (2020)
12. T.N. Weerasinghe, I.A.M. Balapuwaduge, F. Y. Li, Preamble reservation based access for grouped mMTC devices with URLLC requirements, in *Proceeding IEEE International Conference Communications (ICC)* (Shanghai, 2019)
13. Z. Shi, X. Xie, H. Lu, Deep reinforcement learning based intelligent user selection in massive MIMO underlay cognitive radios. IEEE Access **7**, 110,884–110,894 (2019)
14. P. Popovski et al., Wireless access in ultra-reliable low-latency communication (URLLC). IEEE Trans. Commun. **67**(8), 5783–5801 (2019)
15. L. Zhao, X. Chi, L. Qian, W. Chen, Analysis on latency-bounded reliability for adaptive grant-free access with multi-packets reception (MPR) in URLLCs. IEEE Commun. Lett. **23**(5), 892–895 (2019)

16. S. Doan, A. Tusha, H. Arslan, NOMA with index modulation for uplink URLLC through grant-free access. IEEE J. Sel. Top. Sign. Proces. **13**(6), 1249–1257 (2019)
17. Y. Jiang, Network calculus and queueing theory: two sides of one coin, in *Proceeding International ICST Conference on Performance Evaluation Methodologies and Tools* (Brussels, 2009), pp. 1–11
18. O. Naparstek, K. Cohen, Deep multi-user reinforcement learning for distributed dynamic spectrum access. IEEE Trans. Wirel. Commun. **18**(1), 310–323 (2019)
19. L. Liang, H. Ye, G.Y. Li, Spectrum sharing in vehicular networks based on multi-agent reinforcement learning. IEEE J. Sel. Areas Commun. **37**(10), 2282–2292 (2019)
20. Y. Hua, R. Li, Z. Zhao, X. Chen, H. Zhang, GAN-powered deep distributional reinforcement learning for resource management in network slicing. IEEE J. Sel. Areas Commun. **38**(2), 334–349 (2020)
21. Y. Yu, T. Wang, S.C. Liew, Deep-reinforcement learning multiple access for heterogeneous wireless networks. IEEE J. Sel. Areas Commun. **37**(6), 1277–1290 (2019)
22. F. Shah-Mohammadi, A. Kwasinski, Deep reinforcement learning approach to QoE-driven resource allocation for spectrum underlay in cognitive radio networks, in *Proceeding IEEE International Conference on Communications (ICC) Workshops* (Kansas City, 2018)
23. Y. Han, B.D. Rao, J. Lee, Massive uncoordinated access with massive MIMO: a dictionary learning approach. IEEE Trans. Wirel. Commun. **19**(2), 1320–1332 (2020)
24. Z. Chen, F. Sohrabi, W. Yu, Multi-cell sparse activity detection for massive random access: massive MIMO versus cooperative MIMO. IEEE Trans. Wirel. Commun. **18**(8), 4060–4074 (2019)
25. H. Zhang, Y. Liao, L. Song, D2D-U: device-to-device communications in unlicensed bands for 5G system. IEEE Trans. Wirel. Commun. **16**(6), 3507–3519 (2017)
26. M. Gharbieh, A. Bader, H. ElSawy, H. Yang, M. Alouini, A. Adinoyi, Self-organized scheduling request for uplink 5G networks: a D2D clustering approach. IEEE Trans. Commun. **67**(2), 1197–1209 (2019)
27. L. Liu, W. Yu, A D2D-based protocol for ultra-reliable wireless communications for industrial automation. IEEE Trans. Wirel. Commun. **17**(8), 5045–5058 (2018)
28. H. Yang, X. Xie, M. Kadoch, Intelligent resource management based on reinforcement learning for ultra-reliable and low-latency IoV communication networks. IEEE Trans. Veh. Technol. **68**(5), 4157–4169 (2019)
29. S. Han et al., Energy efficient secure computation offloading in NOMA-based mMTC networks for IoT. IEEE Internet Things J. **6**(3), 5674–5690 (2019)
30. R. Mahapatra, Y. Nijsure, G. Kaddoum, N.U. Hassan, C. Yuen, Energy efficiency tradeoff mechanism towards wireless green communication: a survey. IEEE Commun. Surv. Tuts. **18**(1), 686–705 (2016)
31. D. Zhai, R. Zhang, L. Cai, B. Li, Y. Jiang, Energy-efficient user scheduling and power allocation for NOMA-based wireless networks with massive IoT devices. IEEE Internet Things J. **5**(3), 1857–1868 (2018)
32. G. Miao, A. Azari, T. Hwang, E^2 -MAC: energy efficient medium access for massive M2M communications. IEEE Trans. Commun. **64**(11), 4720–4735 (2016)
33. Y. Teng, M. Yan, D. Liu, Z. Han, M. Song, Distributed learning solution for uplink traffic control in energy harvesting massive machine-type communications. IEEE Wirel. Commun. Lett. **9**(4), 485–489 (2020)
34. T. Park, W. Saad, Distributed learning for low latency machine type communication in a massive Internet of Things. IEEE Internet Things J. **6**(3), 5562–5576 (2019)
35. Y. Ruan, W. Wang, Z. Zhang, V.K.N. Lau, Delay-aware massive random access for machine-type communications via hierarchical stochastic learning, in *Proceeding International Conference on Communications (ICC)* (Paris, 2017)
36. C. Di, B. Zhang, Q. Liang, S. Li, Y. Guo, Learning automata-based access class barring scheme for massive random access in machine-to-machine communications. IEEE Internet Things J. **6**(4), 6007–6017 (2019)

37. H. Chang, H. Song, Y. Yi, J. Zhang, H. He, L. Liu, Distributive dynamic spectrum access through deep reinforcement learning: a reservoir computing-based approach. IEEE Internet Things J. **6**(2), 1938–1948 (2019)
38. F. Shah-Mohammadi, A. Kwasinski, Deep reinforcement learning approach to QoE-driven resource allocation for spectrum underlay in cognitive radio networks, in Proceeding IEEE International Conference on Communications (ICC) Workshops (Kansas, 2018)
39. H. Yang, X. Xie, M. Kadoch, Machine learning techniques and a case study for intelligent wireless networks. IEEE Netw. **34**(3), 208–215 (2020)
40. A.M. Koushik, F. Hu, S. Kumar, Intelligent spectrum management based on transfer actor-critic learning for rateless transmissions in cognitive radio networks. IEEE Trans. Mob. Comput. **17**(5), 1204–1215 (2018)

Chapter 5
Learning-Based Maritime Location Privacy Protection

In maritime networks (MNs), ships and other maritime mobile devices release their geographical and semantic information of the visited places (e.g., harbors, passenger terminals, and oil terminals) to request location-based services (LBS). This sensitive information enables the inference attacker to exploit the ship users' identity or business information and thus pose privacy threats to the ship users. Therefore, LBS in MNs must protect the privacy of ship users and address the threat of sensitive location exposure during LBS requests. This chapter presents an RL-based sensitive semantic location privacy protection scheme. This scheme uses the idea of differential privacy to randomize the released ship locations and adaptively selects the perturbation policy based on the sensitivity of the semantic location and the attack history. This scheme enables a ship to optimize the perturbation policy in terms of the privacy and QoS loss without being aware of the current inference attack model in a dynamic privacy protection process. To solve the location protection problem with high-dimensional and continuous-valued perturbation policy variables, we develop a deep deterministic policy gradient (DDPG)-based semantic location perturbation scheme. The actor part is used to generate a continuous privacy budget and a perturbation angle, and the critic part is used to estimate the performance of the policy. This scheme can increase the privacy of ships and other maritime mobile devices, which reduces QoS loss and increases the corresponding utility.

5.1 Introduction

In this part, we first briefly introduce maritime LBS and privacy issues. After that, we summarize the inference attack types and popular location privacy protection methods.

L. Xiao et al., *Reinforcement Learning for Maritime Communications*, Wireless
Networks, https://doi.org/10.1007/978-3-031-32138-2_5

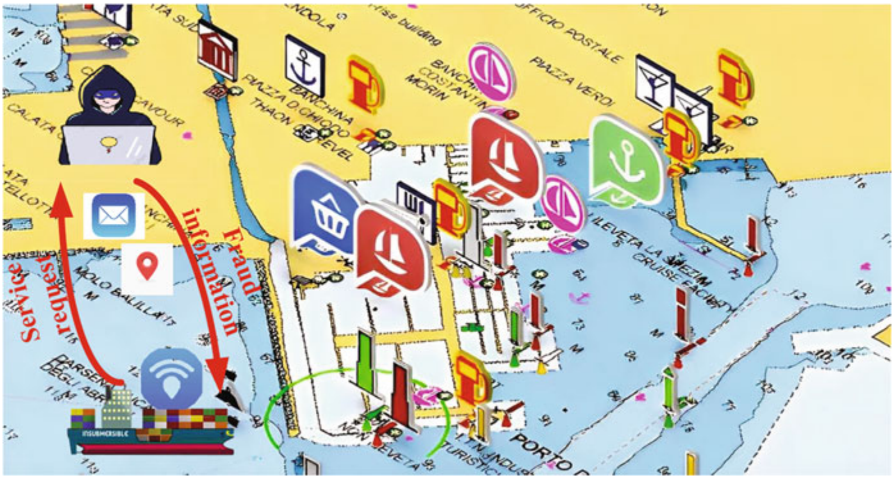


Fig. 5.1 Marine location privacy leakage in LBS

5.1.1 Maritime Location-Based Services and Location Privacy

In the 5/6G era, with the maturity of the Internet of Things, artificial intelligence, blockchain, and other technologies, marine economy, and marine informatization have developed rapidly [1–3]. In particular, the MNs will become essential to the next generation of maritime safety information systems [4]. It can provide all-weather, automatic for all kinds of marine equipment, multidimensional and preciseness maritime location service systems, and a sufficient guarantee of timely ocean rescue, environmental protection, ecological analysis, resource utilization, and other maritime works.

LBSs are significant in MNs, such as maritime navigation, weather forecast, and point of interest recommendations based on mobile users' real-time maritime locations. However, there are many privacy threats to real-time maritime locations. LBS in MNs must protect ship users' privacy and address the threat of the exposure of sensitive locations during LBS requests. Ship users release geographical and semantic information about the visited places (e.g., harbors, passenger terminals, and oil terminals). This sensitive information enables attackers to infer the ship users' identity and business information to achieve fraud, luring improper consumption, and other purposes. As shown in Fig. 5.1, the ship uses terminal equipment to communicate with the server in the MN system to conduct real-time and fast location searches to obtain the corresponding location service. In requesting service, the ship user sends the location and other sensitive information to the LBS server and obtains the corresponding service mark. At the same time, attackers may infer the ship's identity type and expected docking location based on this information and then send fraudulent information. Therefore, location privacy based on location service has become an issue in the marine information environment [5, 6].

Table 5.1 Comparison of privacy protection and their corresponding maritime applications

Types of privacy protection	Techniques	Applications	Potential maritime applications
Data privacy	Randomization	The financial industry	National information security
	Data encryption	Internet domain	Maritime safety
	Differential privacy	Intelligent medical	
	Zero-knowledge proof	Government field	
Identity privacy	Pseudonymization	Social life	Ship type protection
	K-anonymity	Personal information	Crew information protection
	Mix-network	Medical information	
	Blind signature	Career information	
	Group signature		
	Ring signature		
Location privacy	Obfuscation	Travel planning	Dock information
	Differential location privacy	The weather service	The ship trajectory
	Privacy-preserving	Service response	
	Location matching		

To protect maritime mobile users' location and information privacy and deal with the growing privacy leakage problem, scholars have studied the adaptability of various privacy protection technologies in different scenarios, such as data privacy protection, identity privacy protection, and location privacy protection. Table 5.1 lists different scenarios' privacy types and potential maritime communication applications. The relevant privacy protection technologies are listed for each privacy type scenario, and several relevant examples are given.

5.1.2 Inference Attacks

In the context of location privacy protection for maritime mobile terminal requirements, we analyze the advantages and disadvantages of various technologies and focus on application scenarios to find the adaptability of different scenarios. From the perspective of privacy, location information in LBS includes user identity, spatial information (location), and time information, as shown in Fig. 5.2. Each attribute has a different form.

Identity Information An identity is a user's name, E-mail address, or any characteristic that distinguishes one person from another. In LBS, identities can be consistent or inconsistent [7].

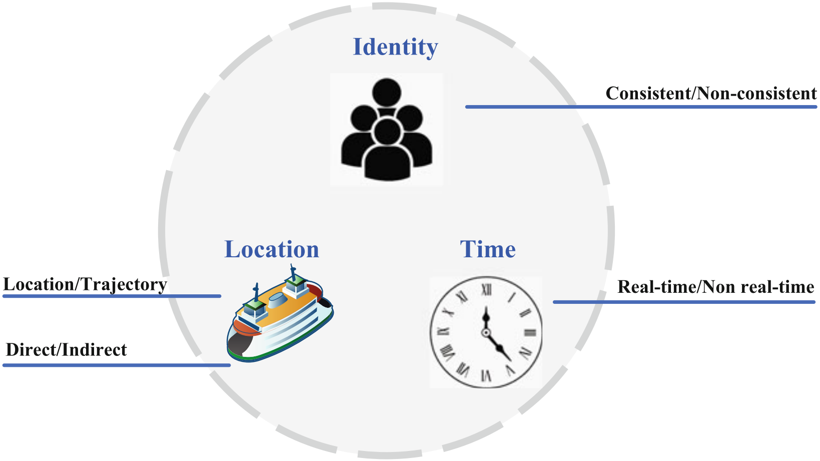


Fig. 5.2 User attribute information

Spatial Information (Location) Spatial information is the primary means of determining location. A location can be described either as a set of coordinates (e.g., longitude and latitude) or as some other forms of information that can be linked to a location, such as a port name. Different types of spatial information can be roughly divided into two categories:

1. Individual locations are scattered and independent of other locations.
2. A trajectory is a set of positions with strong correlation, for example, a ship's trajectory.

Time Information By associating the time stamp with the location, the real-time navigation of the ship is highly dynamic and uncertain.

The attacker's goal is to collect the location information to profit. Figure 5.3 illustrates how the attacker obtains the information, launches the attack, and achieves its goals. The attacker can obtain maritime mobile users' information in the following ways:

1. Collect the location information and historical statistics of shared or published ships.
2. Eavesdropping on maritime networks (communication channels) can expose data traffic transmitted between offshore base stations and maritime mobile terminals.
3. The location information is reflected by the ship task offloading assignment strategy.

The attacker can infer the user's location information from environmental interactions or information provided by the application service provider after processing (context information). Examples of contextual knowledge include: (1) The number of ships staying in a specific area within a particular time; (2) Relationship between

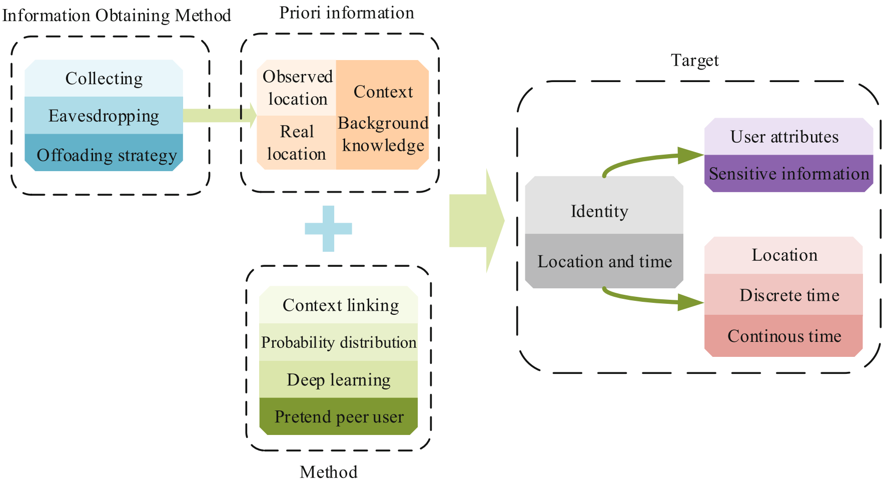


Fig. 5.3 Attacker's pattern flow in the user's privacy leakage

different types of ships; (3) Correlation between ship type and its reported location; (4) Restrictions on the passage of ships in different sea lanes.

Attack Types (1) Identity attack: The presumed location information is used to confirm the user's identity. For example, when docking at the passenger terminal, it is inferred that the ship type is a passenger ship, and when entering the oil terminal, it is judged to be an oil tanker. (2) Location attack: It mainly identifies sensitive areas, such as oil stations and wharves. The serious threat will also determine the specific arrival time of the ship or combine part of the inference event sequence to form a trajectory tracking attack. (3) Semantic attack: When a ship requests service, it may publish some semantic information at the same time, including the attributes of the current location, such as different port types and different sea areas, which can expose more sensitive information about the ship.

Attack Methods (1) Combined inference of context and background. The context and background knowledge obtained by the attacker strongly correlate with the location information observed to infer the ship user's actual location. In a specific environment, the guessing area can be reduced to the specified area, and then all irrelevant areas can be eliminated to infer the ship's location. For example, suppose the attacker knows the departure point of a ship and finds the departure point in the registration information list of a dock. In that case, it can speculate that the ship has arrived at the dock in a certain period and thus obtained relevant information such as the ship's type. (2) Probability-based attack. By observing the location, the attacker can get the posterior distribution of the corresponding location to realize Bayesian attack inference and optimal inference. (3) In the inference attacks using deep learning, the transition matrix is obtained from the training data and applied with the localization attack to infer the user's actual location at a specific time from the disturbed trajectory [8]. (4) Attackers can use semantic information to improve the accuracy of inferring users' sensitive information [9].

5.1.3 Location Privacy Protection

Many studies have discussed mobile users' location privacy protection, which generally has the following forms:

(1) Encryption mechanism: The main problems of location protection mechanisms based on encryption are the computational complexity and the requirement of a cooperative server. Some location encryption techniques will completely hide the user's location information and lose the utility of location service [10].
(2) Anonymization mechanism: k-anonymity achieves location privacy protection through the generalization and suppression algorithm and hides in the nearest $k-1$ location to the user [11]. But k-anonymity is no longer robust to an attacker with background knowledge and relies on trusted third-party servers. Users keep location privacy by changing user names or pseudonyms within the hybrid area. For example, a dynamic hybrid region is studied in [12] to protect the location privacy in the maritime communication network, which is dynamically formed when the maritime device requests LBS.
(3) Perturbation mechanism: The perturbation mechanism spoofs the attacker by adding noise to perturb the actual location to a fake location. Since the generation of fake locations is randomly selected by the user's mobile terminal, this location privacy protection mechanism can achieve a good level of location privacy without any trusted server. Geo-indistinguishability [13] formally defines the notion of protecting a user's location within a radius r, i.e., the privacy protection level, which is achieved by adding controlled random noise. Specific definitions are described as follows: For any given location x and y, the perturbation mechanism $\mathscr{M}$ satisfies ϵ-geo-indistinguishability if and only if the following inequality holds:

$$\frac{\Pr\left(\mathscr{M}(x'|x)\right)}{\Pr\left(\mathscr{M}(x'|y)\right)} \leq e^{d(x,y)\epsilon}. \tag{5.1}$$

This definition states that mechanism $\mathscr{M}$ makes the probability distribution of the perturbed location x' based on the real location x and y similar. The degree of similarity is the difference between the probability distribution of the perturbed location generated by two real locations, which is determined by the privacy budget ϵ and the Euclidean distance $d(x, y)$. Moreover, this definition states that all locations within a given circle are indistinguishable from the attacker's point of view. Therefore, this mechanism can ensure that even if the attacker already knows that the user is in a given range, the accuracy of locating the user's actual location cannot be increased by observing the perturbed location x'.

(4) Semantic location privacy protection: Semantic location refers to the geographical area with the same attributes, such as passenger port, cargo port, etc. Semantic location is considered to be another dimension of users' location information. Therefore, the information can expose more sensitive users' private information to an attacker. In addition, the attacker can infer the voyage of the ship model by observing the time correlation between different semantic locations. For example, at a specific time point, the ship goes to the container piled wharf after leaving the oil terminal. A semantic-aware location privacy protection mechanism is studied in [14], which protects the user's location and the corresponding semantic location information. This mechanism needs to accurately classify different semantic locations and protect their specific semantic location information by publishing upper-level semantic label categories, because the semantic label category at the next level obsesses the current specific semantic information. The mechanism designed in [15] can reduce the loss of LBS service quality by adjusting the disturbance level of geographic location and semantic location. However, this mechanism does not consider the specific attack model.

The above description shows that different basic notions of location privacy protection schemes have different protection effects and objectives. Encryption schemes can reduce the risk of an attacker gaining access to information; anonymization destroys the link between identity and place, rendering anonymous information worthless; the location perturbation mechanism can blur the location information and reduce the risk of location information exposure. In addition, each method needs to consider different types of attackers. The perturbation scheme focuses on spatial and temporal information, while anonymization emphasizes identity protection; cryptography protects all three attributes of location information. Semantic location protection is used to protect sensitive location information.

Currently, most location protection mechanisms ignore the semantic information of the location. However, the attacker can analyze the location semantics of the query requests processed by the location privacy protection scheme through background knowledge. Most of the existing semantic privacy protection schemes consider locations with different social functions as locations with different semantics and classify location semantics into medical treatment, education, dining, travel, and other categories. Simply generating false location sets with multiple semantic types is not enough to defend against the attacker with background knowledge, and it cannot dynamically protect locations according to the user requirements and the sensitivity of different locations. Therefore, this chapter proposes a sensitive semantic location privacy protection scheme based on reinforcement learning (RL). The proposed scheme uses the idea of differential privacy to randomize the published ship locations and adaptively selects the perturbation strategy according to the semantic location sensitivity and attack history.

5.2 Related Work

LBS in MNs, such as the real-time transportation information report, the dock status information of different terminals, and point of interest (POI) recommendations, significantly improve the efficiency of maritime applications [16–18]. However, the location data that indicates the ship user's identity and business information can be leaked from the LBS server or eavesdropped by attackers during the information exchange process for monetary or malicious purposes. MNs must protect users' location privacy against attackers who can infer the users' location and private information from the obtained location data [5, 19–22].

The contextual information attached to the location data exposes more of the private information of users to the attacker. To represent the contextual information the location coordinates can reveal, we use semantic location to describe a region with the same attribute, e.g., passenger terminal, cargo terminal, oil terminal, and fisherman's wharf [21]. Different semantic locations may have different sensitivity levels and demand different privacy protections. For example, the oil terminal may be more sensitive than the passenger terminal, which needs more effort to hide this information from the attacker. However, most location protection schemes have overlooked the semantic location as an additional dimension of the location data.

For example, a user requests the service from an LBS server to obtain the nearest POI recommendation when he/she is in an oil terminal. However, the oil terminal is considered as a sensitive location by the user and does not want to be identified by the LBS server. The semantic locations have strong correlations that can provide extensive information for the attacker to infer more accurately. Then the attacker can use location semantics to strengthen attacks on users' locations and privacy information. For instance, the attacker can infer that the above users' business may be related to oil transactions. Then the advertisement attacker may send semantic-related spams or scams frequently to the user based on his/her current semantic information, degrading the LBS experience of the user [23]. When semantic locations are disclosed, the users' privacy level drops considerably. As a consequence, despite the convenience LBS can bring, many users will be unwilling to utilize LBS services when their location privacy is at risk, which will also impede the success of the LBS.

To protect the user's location privacy in LBS, a location perturbation scheme generalizes the well-known concept of differential privacy (DP) [24] with geo-indistinguishability to enhance LBS applications by using a planar Laplacian mechanism to generate an approximate location[13]. However, this scheme considers the uniform privacy demands and ignores the semantic locations with different sensitivities and usually overestimates/underestimates the user's privacy. For example, assuming a ship is docked at an oil terminal. Suppose the scheme discloses the ship's location that is very near to the actual location of an oil terminal. In that case, the attacker can still infer that the user is in the same oil terminal with a high probability. Suppose a passenger terminal and a cargo terminal are located at the ship's west and east, respectively, at the same distance. In that case, the cargo terminal is more sensitive to the user than the passenger terminal. A traditional location perturbation scheme might release a location in the cargo terminal to protect the current location privacy. Still, the presented perturbation mechanism with the awareness of the location semantic's sensitivity tends to release a perturbed location with less sensitivity in the passenger terminal.

Location protection against semantic attack based on k-anonymity and l-diversity is studied in [11]; however, this scheme usually needs a trusted third party and costs high computation resources to generate the cloak area. A dynamic differential location privacy with personalized demands of a user is studied in [25]. The two-phase location perturbation that combines the geo-indistinguishability and expected inference error [26] is constructed to protect location privacy based on the user's demands by adding more noise to susceptible locations.

A minimax learning algorithm in [27] is studied to defend against advanced attackers and protect the sensor-equipped smartphones' context privacy. An improved Q-learning algorithm is investigated in [28] to help the user make better decisions concerning location disclosure and make a trade-off between the users' privacy and recommendation quality. Q-learning is applied in [29] to reduce the QoS loss while perturbing the location data to protect the semantic location privacy against the inference attacker, in which the ship just perturbs the location data to make sure that the attacker cannot infer the current semantic location.

In this chapter, we study the semantic location privacy protection scheme by incorporating the semantic locations' sensitivity into the location perturbation scheme to induce the attacker's inference resulting in less sensitive semantic locations. Instead of just adding more extensive noise to protect susceptible semantic locations, which will degrade QoS a lot [25], the investigated scheme tends to release a less sensitive one instead of a susceptible one to improve both privacy and QoS. In this way, the current and the potential semantic location can be protected better with the released less sensitive semantic location. In the long-term view, this perturbation scheme can also protect the ship's trajectory and avoid the semantic correlated inference of the attacker.

Most recent location privacy protection research relies on a trusted third party, pays little attention to the semantic sensitivity, and depends on a specific attack model. However, attackers may dynamically change their attack policy based on the network and service state, aiming to infer the user's semantic location in dynamic MNs. Besides, it is difficult to pre-determine the privacy requirement for dynamically changing semantic locations with different sensitivities and LBS requests of the ship user. Moreover, it is noted that the future system state of the ship only depends on the current state and location perturbation policy and does not depend on previous perturbation history. Thus, the location perturbation process of the ship can be modeled by an MDP. Besides, the user's environment is dynamically changing, and the accurate inference attack model is hard to obtain. RL techniques are applied to explore the optimal policy by trial and error with sufficient interactions without knowing the environmental parameters. Consequently, we can apply RL techniques, such as Q-learning, to derive the optimal perturbation policy without being aware of the inference attack model.

We first present a reinforcement learning-based semantic location perturbation scheme (RSLP) for MNs with a differential privacy technique against inference attackers. This scheme dynamically chooses the privacy budget to randomize the released locations to protect the sensitive semantic locations. Notice that releasing the perturbed locations reduces the QoS in location-based service. We use RL [30, 31] to choose the perturbation policy based on the current state consisting of the current semantic location, the location sensitivity, and the attack history [27] to dynamically balance the QoS loss and privacy.

However, the RSLP-based scheme can only solve the location protection problems by discretizing the continuous perturbation policies into a finite set of discrete levels. This method introduces the quantization of the privacy budget and perturbation angle that destroys the completeness of the continuous space and removes some vital information. If the quantization of the perturbation policy is not suitable, the scheme may derive a policy that is not actually optimal. Besides, the RSLP faces the curse of dimensionality, which means that the sizeable action-state space will decrease the learning speed and degrade the semantic location privacy protection performance. The deep deterministic policy gradient (DDPG) algorithm developed in [32] is effective for problems with the continuous-valued location privacy protection policy, which can be used by the ship to explore the optimal perturbation policy in a continuous space.

Thus, a DDPG-based semantic location perturbation scheme (DSLP) is developed to solve the location privacy protection problem with a continuous-valued perturbation policy. In this scheme, the actor part generates a continuous policy of privacy budget and perturbation angle, and the critic part estimates the location privacy protection performance of the perturbation policy. Based on the analysis of the paper [31], the IoT platforms such as Nvidia Tegra K1 and Qualcomm Snapdragon 800, which support deep learning, can run the presented DSLP-based scheme [33]. This indicates that the presented learning-based semantic location protection schemes can be implemented into the real MN environment. More specifically, the significant contribution of this chapter is summarized as follows:

1. We formulate a location privacy-preserving framework that considers not only the geographical location but also the semantic dimension of the location against the honest-but-curious attackers in LBS. The ship can protect its highly sensitive semantic locations locally without a trusted third party and avoid the semantic correlated inference of the attacker.
2. We present an RL-based semantic location perturbation scheme and use the idea of the geo-indistinguishability to adaptively select an appropriate privacy budget and perturbation angle to reduce the exposure of highly sensitive locations.
3. A DDPG-based semantic location perturbation scheme that can automatically balance the trade-off between the privacy and QoS loss with high efficiency is investigated to select the perturbation policy from a continuous-valued perturbation policy set without knowledge of the inference attack model.

The remainder of this chapter is organized as follows. Section 5.3 presents the system model. Then, an RSLP- and a DSLP-based semantic location perturbation schemes are presented in Sect. 5.4. The evaluation results are provided in Sect. 5.5, and this work is concluded in Sect. 5.6.

5.3 System Model

We consider a ship equipped with a Global Positioning System (GPS) which has localization capabilities. As shown in Fig. 5.4, the ship moves in an offshore area and requests the LBS from the server via a coast base station [34, 35]. To get LBS such as real-time transportation information reports, the dock status information of different terminals in the coastal port, and POI recommendations, the ship sends its request to the LBS server together with the perturbed locations. The server is honest-but-curious and tries to infer the ship's true geographical locations and the corresponding semantic types by observing the obtained location data. Then the server may send semantic information-related spam or scams to users to gain commercial profit.

The GPS-equipped ship moves in a given offshore area $\mathscr{D}$ at time slot k, which is divided into N nonoverlapping cells $\boldsymbol{d}^{(k)}$, representing the coordinates of different locations, i.e., geo-location. The semantic location visited by the ship is denoted by $c^{(k)} \in [C_i]_{0\leq i\leq N}$, in which the tags represent the semantic of different regions, e.g., C_1 passenger terminal and C_2 represents cargo terminal, as shown in Fig. 5.4. Different semantic locations in this geographical area may own different sizes, and

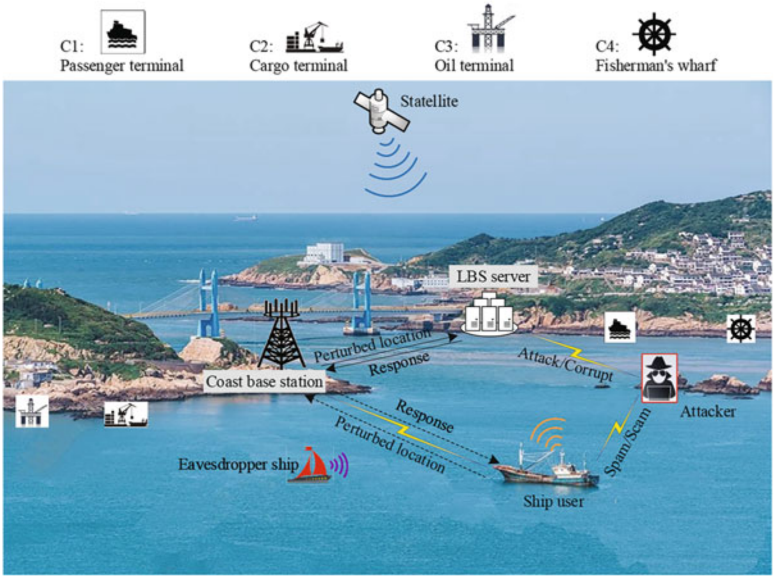


Fig. 5.4 Illustration of the sensitive semantic location privacy protection scheme

the semantic location with a larger size consists of more cells. The sensitivity of the corresponding semantic location at time slot k is represented by $l^{(k)} \in [L_i]_{0 \le i \le 3}$. The sensitivity level of a semantic location represents the importance of the semantic location for the user privacy, i.e., the higher the sensitivity level a semantic location has, the stronger demand for hiding the true semantic location of the user is.

5.3.1 Network Model

The ship's transition between various semantic locations is captured by a Markov model according to [36], which states that

$$\Pr[c^{(k)} = C_i | c^{(1)}, ..., c^{(k-1)}] = \Pr[c^{(k)} = C_i | c^{(k-1)}]. \tag{5.2}$$

Discrete time slots over a limited time $k \in \{1, 2, ...\}$ are considered.

The ship sends its location to the LBS server to request services. Due to privacy concerns, the real location has to be perturbed before sending it out. The idea of the generalization version of DP, geo-indistinguishability, is adopted to perturb the ship's location [13]. The perturbation policy $\boldsymbol{a}^{(k)}$ consists of the privacy budget $x^{(k)}$ and the perturbation angle $\vartheta^{(k)}$ considering the semantic constraint, i.e., $\boldsymbol{a}^{(k)} = [x^{(k)}, \vartheta^{(k)}] \in \mathbf{A}$, in which $\mathbf{A}$ is the possible perturbation policy set. The perturbation policy is selected to decrease the perturbation distance as much as possible with a less sensitive semantic location, to increase the privacy and reduce the QoS loss. On the one hand, as shown in Fig. 5.4, with the same perturbation distance $r^{(k)}$, the ship tends to release a location in the passenger terminal with less sensitivity rather than the oil terminal. On the other hand, if there are several less sensitive locations surrounding the current actual location, the ship tends to release a less sensitive location nearer its actual location. In this way, the ship can improve its privacy by reducing the exposure of highly sensitive locations and reduce the QoS loss. The ship generates a perturbed location $\tilde{\boldsymbol{d}}^{(k)}$ based on the selected perturbation policy; meanwhile the perturbed semantic location $\tilde{c}^{(k)}$ can be obtained according to the map.

Similar to [14], at a given time slot, the ship's perturbed location is independent of the other time slots. The ship has different privacy demands while visiting different semantic locations with different sensitivities, and the ship's tolerance for the QoS loss is different when it asks for different kinds of LBSs. Thus, the ship has to adjust its perturbation policy adaptively as the sensitivity of the visited semantic location and its QoS loss tolerance change. The ship can estimate its privacy by the spam/scams it receives and evaluate the QoS loss based on its service experience.

5.3.2 *Attack Model*

The attacker is considered as an honest-but-curious LBS server or an external attacker (e.g., eavesdropper ship) who can observe the output of the location protection scheme. Its main purpose is to locate the ship at a given time or identify the semantic location type that the ship visits. We consider the attacker owns some prior knowledge about the ship and uses the observed locations to infer the ship's location $\hat{\boldsymbol{d}}^{(k)}$ [37]. Even if the ship perturbs its location independently in each time slot, the attacker assumes that there are correlations between the locations of a ship and therefore models the ship's mobility to infer the ship user's identity and ship's business. The attacker is assumed to know the maritime map that indicates the semantic locations corresponding to their geo-locations and their cover areas, and it infers the semantic of the current location $\hat{c}^{(k)}$ based on $\hat{\boldsymbol{d}}^{(k)}$.

The actual location of the ship is known only by itself, while the attacker can infer the business or other private information of the user according to the information reflected by the users' geographical locations and the corresponding semantic locations. Then the attacker sends scam or spams to the user based on the inference results of the ship's private information.

5.3.3 *Privacy Protection Problem*

The ship can evaluate the privacy $p^{(k)}$ by observing the received scams or spams similar to [14, 27] to estimate the difference between the sensitivity of the inferred location by the attacker $\hat{l}^{(k)}$ and that of the actual location $l^{(k)}$. The QoS loss $q^{(k)}$ is evaluated based on the Euclidean distance between its perturbed geo-location $\tilde{\boldsymbol{d}}^{(k)}$ and actual geo-location $\boldsymbol{d}^{(k)}$ according to [37], which can reflect the quality of the user's experience. The ship improves its location privacy by randomizing its released location to confuse the attacker. This also leads to the QoS loss because the ship uses the fake location to request the LBS. Thus, our work focuses on balancing the trade-off between the privacy and QoS loss by adjusting the location perturbation policy. For ease of reference, our commonly used notions are summarized in Table 5.2.

5.4 Semantic Location Privacy Protection

The ship's next state only depends on the current state and perturbation policy. It has nothing to do with the past location protection history. Thus, the ship's location privacy protection process can be viewed as an MDP. In this section, we can apply RL techniques, such as Q-learning and DDPG, to derive the optimal perturbation policy without being aware of the inference attack model.

Table 5.2 List of notations

Symbol	Description
$x^{(k)} \in [\check{X}, \hat{X}]$	Privacy budget
$\vartheta^{(k)}$	Perturbation angle
ϱ	Weighting parameter of privacy
ν/κ	Parameters determining the tolerance of the QoS loss
$\boldsymbol{d}^{(k)}/\tilde{\boldsymbol{d}}^{(k)}/\hat{\boldsymbol{d}}^{(k)}$	Actual/perturbed/inferred geo-location
$c^{(k)}/\tilde{c}^{(k)}/\hat{c}^{(k)}$	Actual/perturbed/inferred semantic location
$l^{(k)}/\tilde{l}^{(k)}/\hat{l}^{(k)}$	Actual/perturbed/inferred semantic sensitivity level
$r^{(k)} \in [\hat{R}, \check{R}]$	Perturbed distance between $\boldsymbol{d}^{(k)}$ and $\tilde{\boldsymbol{d}}^{(k)}$
$p^{(k)}$	Privacy
$q^{(k)}$	QoS loss

5.4.1 *Learning-Based Semantic Location Perturbation*

In this section, we present an RL-based semantic location perturbation scheme (RSLP) that enables a ship to act as a learning agent to optimize its perturbation policy in the dynamic location privacy protection process against attackers without knowledge of the inference attack model. This scheme derives the optimal perturbation policy via trial and error based on the current state $\boldsymbol{s}^{(k)}$ consisting of the location of the ship, the sensitivity level of the semantic location, and the estimated attack strength history.

At time slot k, the ship observes its geo-location $\boldsymbol{d}^{(k)}$, the corresponding semantic location $c^{(k)}$, and its sensitivity level $l^{(k)}$. The ship estimates its previous semantic location leakage with $\varpi^{(k-1)}$ based on the degree of correlation between the received scams or spams and the previous semantic locations, similar to [27], which formulate the current state $\boldsymbol{s}^{(k)} = [\boldsymbol{d}^{(k)}, c^{(k)}, l^{(k)}, \varpi^{(k-1)}]$.

Let parameter x denote the privacy budget, and $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ are any two locations in possible regions $\mathscr{D}$. We use the idea of the x-geo-indistinguishability [13] to determine the perturbed distance in the sensitive semantic location protection. More specifically, the perturbation mechanism transforms any two actual locations to fake locations with a similar probability distribution. The similarity of the probability distributions is determined by the privacy budget x and the selected circular with radius R ($R \geq ||\boldsymbol{d}_1 - \boldsymbol{d}_2||_2, \forall \boldsymbol{d}_1, \boldsymbol{d}_2 \in \mathscr{D}$), with $\|\cdot\|_2$ represents the Euclidean norm. This means that all the locations within the circle with radius R are indistinguishable from the eyes of the attacker. The smaller privacy budget x will make the two distributions that transform any two actual locations to the fake location closer, yielding a higher privacy level.

In the RSLP scheme, the ship selects its action, i.e., perturbation policy $\boldsymbol{a}^{(k)} = [x^{(k)}, \vartheta^{(k)}]$, which is made up of the privacy budget $x^{(k)} \in \{\check{X}, X_1, ..., \hat{X}\}$, where $\check{X}$ is the minimum privacy budget and $\hat{X}$ is the largest privacy budget, and the perturbation angle $\vartheta^{(k)} \in \{0, ..., 2\pi\}$ under the constraint of semantic locations, as shown in Algorithm 5. More specifically, based on the selected $x^{(k)}$, the ship firstly uses a gamma distribution to sample a $r^{(k)}$, i.e., $r^{(k)} \sim \Gamma(2, 1/x^{(k)})$, which represents the distance between $\boldsymbol{d}^{(k)}$ and $\tilde{\boldsymbol{d}}^{(k)}$. Then with $\vartheta^{(k)}$ determining the perturbation angle, the ship calculates the perturbed location $\tilde{\boldsymbol{d}}^{(k)}$ by

$$\tilde{\boldsymbol{d}}^{(k)} = \boldsymbol{d}^{(k)} + \left\langle r^{(k)} \cos \vartheta^{(k)}, r^{(k)} \sin \vartheta^{(k)} \right\rangle. \tag{5.3}$$

The ε-greedy policy is used to choose an action based on $\boldsymbol{s}^{(k)}$ as a trade-off between exploration and exploitation.

In the perturbation policy, the privacy budget $x^{(k)}$ is used to trade off between the privacy and the QoS loss, and the angle $\vartheta^{(k)}$ is used to optimize privacy by controlling the perturbed semantic location's sensitivity to be as low as possible. With the map and $\tilde{\boldsymbol{d}}^{(k)}$, the perturbed semantic location $\tilde{c}^{(k)}$ can be easily obtained. After that, the ship sends a service request together with its perturbed location $(\tilde{\boldsymbol{d}}^{(k)}, \tilde{c}^{(k)})$ to the LBS server. Then the server sends the POI recommendations, maybe together with some scams or spam as feedback to the ship.

The ship then evaluates its privacy $p^{(k)}$ based on the received spams or scams similar to [14] and [27]. That is to say, if they have certain relation with the current semantic location, the location privacy is leaked. The degree of privacy leakage can be represented by the difference between the inferred semantic location's sensitivity $\hat{l}^{(k)}$ and the actual semantic location's sensitivity $l^{(k)}$, i.e., $l^{(k)} - \hat{l}^{(k)}$. Besides, the higher sensitive semantic location located by the ship, the more effort should be made to hide its semantic location to gain higher privacy. Thus, we have

$$p^{(k)} = \left(l^{(k)} - \hat{l}^{(k)}\right)\left(l^{(k)} + \tau\right), \tag{5.4}$$

where the weighting parameter τ is a constant and $\tau \neq 0$.

The QoS loss is influenced by the perturbed distance and the type of LBS applications. According to [26], the QoS loss can be measured by the distance between the actual geo-location and the perturbed geo-location $\| \boldsymbol{d}^{(k)} - \tilde{\boldsymbol{d}}^{(k)} \|_2^2$, which can reflect the quality of the user's experience. We all know that if the perturbed distance is larger, the POI recommendations or feedback results will be less accurate. Besides, we consider the QoS loss of different kinds of LBS applications has different sensitivities to the perturbed distance. For instance, the application of weather forecasts is less sensitive to the perturbed distance. The result will be the same if the perturbed location is within a given marine area. However, the QoS loss of the LBS application of the real-time transportation information report is much more sensitive to the perturbed distance. This feature can be captured by parameters κ and ν. According to [39], if the perturbed distance becomes larger than the threshold κ/ν, the QoS loss will drop faster. And the drop speed is determined by the parameter ν. Thus, the QoS loss is given by

$$q^{(k)} = \arctan\left(\nu \left\|\boldsymbol{d}^{(k)} - \tilde{\boldsymbol{d}}^{(k)}\right\|_2^2 - \kappa\right), \tag{5.5}$$

where ν and κ determine the sensitivity of the QoS loss to the perturbed distance for a given LBS application.

The weighting parameter ϱ represents the importance of the privacy over the QoS loss. Thus, the utility of the ship at time slot k represented by $u^{(k)}$ depends on both the privacy and QoS loss. According to (5.4) and (5.5), we have

$$\begin{aligned} u^{(k)} =& \varrho\left(l^{(k)} - \hat{l}^{(k)}\right)\left(l^{(k)} + \tau\right) \\ & - \arctan\left(\nu \left\|\boldsymbol{d}^{(k)} - \tilde{\boldsymbol{d}}^{(k)}\right\|_2^2 - \kappa\right). \end{aligned} \tag{5.6}$$

When the actual location is far away from nonsensitive ones, based on current LBS applications and current semantic location, if the ship treats the privacy more important than the QoS (i.e., ϱ is large), then the ship tends to select a perturbed location located at a less sensitive area, even if it is far away. If the ship treats the QoS more important than the privacy (i.e., ϱ is small), then the ship tends to select a perturbed location which is not very far away from the actual one, with a less sensitive location as much as possible.

The Q-function $Q(\boldsymbol{s}, \boldsymbol{a})$ is the expected discount long-term reward of a ship that uses the perturbation policy $\boldsymbol{a}^{(k)}$ at state $\boldsymbol{s}^{(k)}$, which is updated according to the iterative Bellman equation as shown in Algorithm 5. We use the transfer learning method in [38] to initialize the Q-values as $\bar{Q}$ with location perturbation experiences in similar MN environments such as a number of typical attack strength of attacker and map type.

5.4.2 *Deep RL-Based Semantic Location Perturbation*

The RSLP-based semantic location perturbation scheme is inefficient for ships in a big map with a large amount of states, and the naive discretization of perturbation policy spaces may cause the ship's failure to find the globally optimal perturbation policy to protect the ship's sensitive semantic locations. To meet the demand of the ship's privacy protection with a practical complicated MN system and high-dimensional continuous perturbation policy space, we develop a DDPG-based semantic location perturbation scheme (DSLP) to explore the policy spaces efficiently and improve the sensitive semantic location protection performance.

As shown in Fig. 5.5, the DSLP architecture consists of the critic network and actor network. They are parameterized by the functions $Q(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta})$ and $u(\boldsymbol{s}|\boldsymbol{\omega})$ with parameters $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$, respectively. The ship uses the actor network to select a perturbation policy from a continuous action space and uses the critic network to evaluate the performance and criticize the perturbation policy selected by the actor

Algorithm 5 RL-based semantic location perturbation

```
Initialize the maritime map, $\mathbf{A}$, $\alpha$, $\delta$, $\varepsilon$ and $\varpi^{(0)}$
Set $Q \leftarrow \bar{Q}$ based on the transfer learning method [38]
for $k = 1, 2, 3, \ldots$ do
    Observe $\boldsymbol{d}^{(k)}$ and $c^{(k)}$ according to the maritime map
    Evaluate the sensitivity level of current location $l^{(k)}$
    $\boldsymbol{s}^{(k)} = [\boldsymbol{d}^{(k)}, c^{(k)}, l^{(k)}, \varpi^{(k-1)}]$
    Select $\boldsymbol{a}^{(k)} = [x^{(k)}, \vartheta^{(k)}]$ according to:
    $\Pr\left(\boldsymbol{a}^{(k)} = \hat{\boldsymbol{a}}\right) = \begin{cases} 1-\varepsilon, & \hat{\boldsymbol{a}} = \arg\max_{\boldsymbol{a}'} Q\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}'\right) \\ \frac{\varepsilon}{|\mathbf{A}|-1}, & \text{o.w} \end{cases}$
    Sample $r^{(k)} \sim \Gamma\left(2, 1/x^{(k)}\right)$
    Calculate $\tilde{\boldsymbol{d}}^{(k)}$ via (5.3)
    Map the $\tilde{\boldsymbol{d}}^{(k)}$ to $\tilde{c}^{(k)}$
    Send $(\tilde{\boldsymbol{d}}^{(k)}, \tilde{c}^{(k)})$ to the LBS server
    Receive the response from the LBS server
    Evaluate the privacy $p^{(k)}$
    Evaluate the QoS loss $q^{(k)}$
    Calculate the utility $u^{(k)}$ via (5.6)
    $Q(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}) \leftarrow (1-\alpha)\, Q\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}\right) + \alpha\left(u^{(k)} + \delta \max_{\boldsymbol{a}^{(k)} \in \mathbf{A}} Q\left(\boldsymbol{s}^{(k+1)}, \boldsymbol{a}^{(k)}\right)\right)$
end for
```

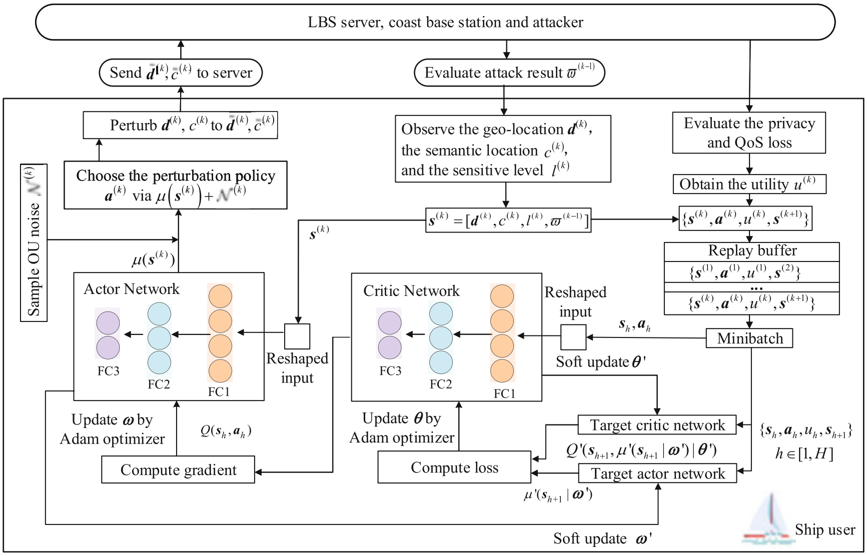

LBS server, coast base station and attacker
Send $\tilde{\boldsymbol{d}}^{(k)}, \tilde{c}^{(k)}$ to server
Evaluate attack result $\varpi^{(k-1)}$
Perturb $\boldsymbol{d}^{(k)}, c^{(k)}$ to $\tilde{\boldsymbol{d}}^{(k)}, \tilde{c}^{(k)}$
Observe the geo-location $\boldsymbol{d}^{(k)}$, the semantic location $c^{(k)}$, and the sensitive level $l^{(k)}$
Evaluate the privacy and QoS loss
Obtain the utility $u^{(k)}$
Sample OU noise $\mathcal{N}^{(k)}$
Choose the perturbation policy $\boldsymbol{a}^{(k)}$ via $\mu\left(\boldsymbol{s}^{(k)}\right)+\mathcal{N}^{(k)}$
$\boldsymbol{s}^{(k)}$
$\boldsymbol{s}^{(k)} = [\boldsymbol{d}^{(k)}, c^{(k)}, l^{(k)}, \varpi^{(k-1)}]$
$\{\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}, u^{(k)}, \boldsymbol{s}^{(k+1)}\}$
Replay buffer
$\{\boldsymbol{s}^{(1)}, \boldsymbol{a}^{(1)}, u^{(1)}, \boldsymbol{s}^{(2)}\}$
...
$\{\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}, u^{(k)}, \boldsymbol{s}^{(k+1)}\}$
$\mu(\boldsymbol{s}^{(k)})$
Actor Network
FC3
FC2
FC1
Reshaped input
Critic Network
FC3
FC2
FC1
Reshaped input
$\boldsymbol{s}_h, \boldsymbol{a}_h$
Minibatch
Soft update $\boldsymbol{\theta}'$
Update $\boldsymbol{\varpi}$ by Adam optimizer
$Q(\boldsymbol{s}_h, \boldsymbol{a}_h)$
Update $\boldsymbol{\theta}$ by Adam optimizer
Target critic network
$Q'(\boldsymbol{s}_{h+1}, \mu'(\boldsymbol{s}_{h+1} | \boldsymbol{\varpi}') | \boldsymbol{\theta}')$
$\{\boldsymbol{s}_h, \boldsymbol{a}_h, u_h, \boldsymbol{s}_{h+1}\}$
$h \in [1, H]$
Compute gradient
Compute loss
Target actor network
$\mu'(\boldsymbol{s}_{h+1} | \boldsymbol{\varpi}')$
Soft update $\boldsymbol{\varpi}'$
Ship user

Fig. 5.5 Illustration of the deep deterministic policy gradient-based semantic location perturbation scheme for the ship user

part. The ship also maintains the target critic and actor networks to calculate the target values for the critic network updating, in which these two target networks output functions $Q'(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta}')$ and $\mu'(\boldsymbol{s}|\boldsymbol{\omega}')$ with parameters $\boldsymbol{\theta}'$ and $\boldsymbol{\omega}'$.

Algorithm 6 DDPG-based semantic location perturbation

```
Initialize the map, $\mathbf{A}$, $\varpi^{(0)}$, $\boldsymbol{\omega}$, $\boldsymbol{\theta}$ and $\mathscr{N}^{(0)}$
for $k = 1, 2, 3, ...$ do
    Observe $\boldsymbol{d}^{(k)}$ and $c^{(k)}$ according to the map
    Evaluate the sensitivity level of current location $l^{(k)}$
    Formulate the current state
    $\boldsymbol{s}^{(k)} = [\boldsymbol{d}^{(k)}, c^{(k)}, l^{(k)}, \varpi^{(k-1)}]$
    Input the current state to the DNN of the actor network
    Select $\boldsymbol{a}^{(k)} = [x^{(k)}, \vartheta^{(k)}]$ according to:
    $\boldsymbol{a}^{(k)} = \mu(\boldsymbol{s}^{(k)}|\boldsymbol{\omega}) + \mathscr{N}^{(k)}$
    Perform the location perturbation and evaluate the performance as Steps 8-15 in Algorithm 5
    Store $\{\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}, u^{(k)}, \boldsymbol{s}^{(k+1)}\}$ in the replay buffer
    Sample minibatch
    $\{\boldsymbol{s}_h, \boldsymbol{a}_h, u_h, \boldsymbol{s}_{h+1}\}, h \in [1, H]$ from the replay buffer
    Update the online networks $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ via (5.7) and (5.8)
    Soft update the target networks $\boldsymbol{\omega}'$ and $\boldsymbol{\theta}'$ via (5.9)
end for
```

More specifically, Algorithm 6 illustrates the detail of DSLP. The ship first initializes the system parameters and then observes the current state $\boldsymbol{s}^{(k)}$ similar to Algorithm 5. Then the ship reshapes the state and inputs it into the deep neural network (DNN) of the actor network with three fully connected (FC) layers. The attack model features and location perturbation details are captured by the DNN.

At time slot k, the ship chooses the perturbation policy $\boldsymbol{a}^{(k)} = [x^{(k)}, \vartheta^{(k)}]$ in a continuous action space, in which $x^{(k)} \in [\check{X}, \hat{X}]$ is the privacy budget and $\vartheta^{(k)} \in [0, 2\pi)$ is the perturbation direction. The actor network selects $\boldsymbol{a}^{(k)}$ by mapping every state to a determined action with function $\mu(\boldsymbol{s}^{(k)}|\boldsymbol{\omega}^{(k)})$. To improve the exploration efficiency, a noise $\mathscr{N}^{(k)}$ sampled from an Ornstein-Uhlenbeck (OU) process [40] is added to $\mu(\boldsymbol{s}^{(k)}|\boldsymbol{\omega}^{(k)})$ to generate temporally correlated exploration in the learning process, i.e., $\boldsymbol{a}^{(k)} = \mu(\boldsymbol{s}^{(k)}|\boldsymbol{\omega}^{(k)}) + \mathscr{N}^{(k)}$. After that, the perturbed location $(\tilde{\boldsymbol{d}}^{(k)}, \tilde{c}^{(k)})$ is generated as in Algorithm 5. The ship then sends $(\tilde{\boldsymbol{d}}^{(k)}, \tilde{c}^{(k)})$ to the LBS server to protect its true location.

After the perturbed location is released to the LBS, the ship evaluates its privacy $p^{(k)}$ and QoS loss $q^{(k)}$ to obtain its utility $u^{(k)}$. The next state is formulated as $\boldsymbol{s}^{(k+1)} = [\boldsymbol{d}^{(k+1)}, c^{(k+1)}, l^{(k+1)}, \varpi^{(k)}]$. In order to learn in minibatch to make efficient perturbation policy optimizations, the experience consisting of the current state $\boldsymbol{s}^{(k)}$, the selected perturbation policy $\boldsymbol{a}^{(k)}$, the obtained utility $u^{(k)}$, and the next state $\boldsymbol{s}^{(k+1)}$, i.e., $\boldsymbol{e}^{(k)} = \{\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}, u^{(k)}, \boldsymbol{s}^{(k+1)}\}$ are sampled from the dynamic MN environment based on the exploration policy and stored in a replay buffer. Thus, the ship can make full use of a set of uncorrelated transitions to explore the optimal perturbation policy. Since the storage space of the replay buffer is finite, the oldest experiences need to be discarded on a rolling basis.

During the updating process of the critic and actor networks, the ship randomly chooses H experiences from the replay buffer and formulates the minibatch, with the h-th experience $\boldsymbol{e}_h = \{\boldsymbol{s}_h, \boldsymbol{a}_h, u_h, \boldsymbol{s}_{h+1}\}, h \in [1, H]$. We use Adam optimizer to update the critic network's weights $\boldsymbol{\theta}$ with the aim of minimizing the following loss function:

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \frac{1}{H} \sum_{h=1}^{H} \Big(u_h + \gamma Q' \left(\boldsymbol{s}_{h+1}, \mu' \left(\boldsymbol{s}_{h+1} | \boldsymbol{\omega}' \right) | \boldsymbol{\theta}' \right) \\ - Q(\boldsymbol{s}_h, \boldsymbol{a}_h \big| \boldsymbol{\theta}) \Big)^2, \tag{5.7}$$

where $\gamma \in [0, 1]$ is the discount factor indicating the ship's uncertainty of the future reward.

The actor network's weight $\boldsymbol{\omega}$ is also updated based on Adam optimizer with respect to the direction of the Q-value gradient as follows:

$$\nabla_{\boldsymbol{\omega}} J \approx \frac{1}{H} \sum_{h=1}^{H} \nabla_{\boldsymbol{a}} Q \left(\boldsymbol{s} = \boldsymbol{s}_h, \boldsymbol{a} = \mu(\boldsymbol{s}_h) \right| \boldsymbol{\theta}) \nabla_{\boldsymbol{\omega}} \mu \left(\boldsymbol{s} = \boldsymbol{s}_h | \boldsymbol{\omega} \right), \tag{5.8}$$

where $\nabla_{\boldsymbol{a}} Q(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta})$ is the Q-function's policy gradient with respect to action $\boldsymbol{a}$. Similarly, $\nabla_{\boldsymbol{\omega}} \mu(\boldsymbol{s}|\boldsymbol{\omega})$ is the actor function's policy gradient with respect to $\boldsymbol{\omega}$.

Instead of directly copying the weights of the critic and actor networks, the ship uses the soft update to keep the output of the target critic and actor networks with parameters $Q'(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta}')$ and $\mu'(\boldsymbol{s}|\boldsymbol{\omega}')$ changing slowly while updating. Thus, the learning stability can be improved. More specifically, with a learning rate $\zeta \ll 1$, the critic and actor networks' learned weights are tracked slowly based on the soft update, updating by

$$\begin{aligned} \boldsymbol{\theta}' &\leftarrow \zeta \boldsymbol{\theta} + (1 - \zeta) \boldsymbol{\theta}', \\ \boldsymbol{\omega}' &\leftarrow \zeta \boldsymbol{\omega} + (1 - \zeta) \boldsymbol{\omega}'. \end{aligned} \tag{5.9}$$

The ship can apply the same neural network to select the location perturbation policy even if the environment, such as the attack model and the map, changes. That's because the presented learning-based location perturbation scheme is a model-free and reinforcement mechanism; thus, the ship can dynamically observe the current MN environment and ship's state and input them into the DNN. The DNN can capture the feature of the variant environment and uses the stored location perturbation experiences to update the network parameters dynamically. In this way, the proposed learning-based location protection scheme can adapt to various attack models and the dynamic MN environments.

5.5 Simulation Results

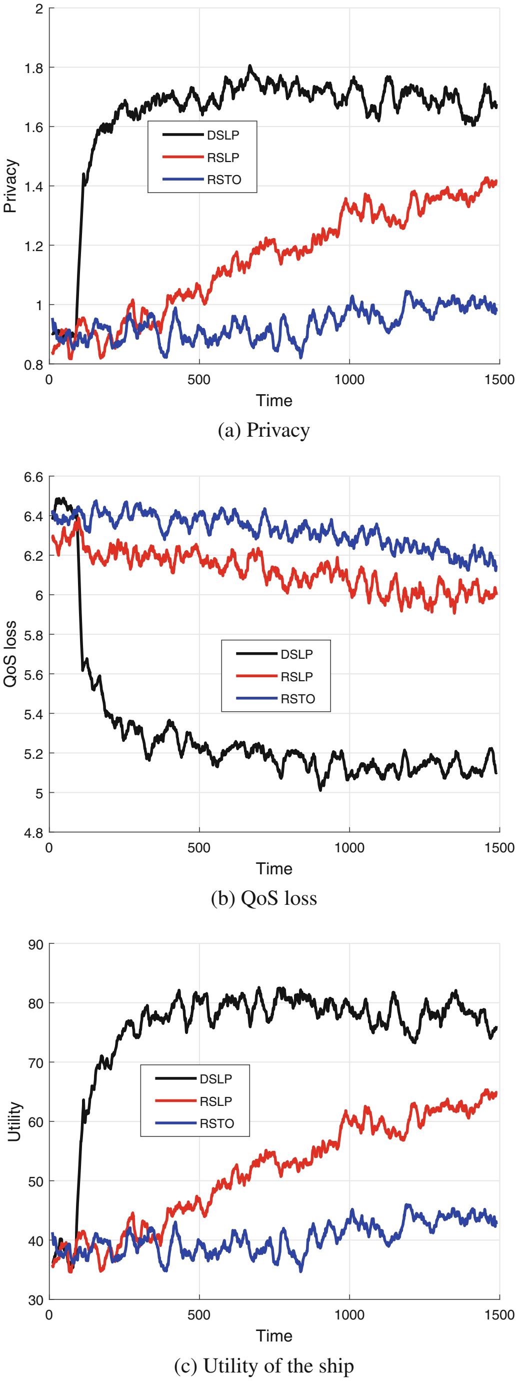
This section evaluates the performance of the presented learning-based semantic location perturbation schemes for MNs. Simulations have been performed for a ship in a square map with equal width and length of 6 km, which is divided into a 10×10 grid with different semantic locations such as passenger terminal, cargo terminal, oil terminal, and fisherman's wharf. The semantic locations in the map belong to four different sensitivity levels from level 0 to level 3. These values are selected to show the difference in the sensitivity of semantic location in the simulation. They can represent different actual values in the real world. The transition of the ship in the grid map is modeled as a Markov chain. According to [41], the privacy budget $x \in (0, 1.4]$, the privacy weight over the QoS loss is set to be 5.

In the simulations of the learning algorithm, the learning rate is set to 0.7, and the discount factor is set to 0.5. The ε-greedy parameter is annealed from 1.0 to 0.1 during the first 500 time slots in the process of learning, and after that ε is fixed to 0.1 for stability. The reinforcement learning-based location obfuscation scheme (RSTO) in [29] neglecting semantic locations with different sensitivities is evaluated as the benchmark scheme. The change of the number of semantic locations, the value of sensitivity levels, or the user's mobility profile can observe a similar trend as the current typical case. Even though the convergence time and convergence value might be different from the current case, they will not impact the advantage of the presented schemes compared with the benchmark.

The performance of the presented RSLP- and DSLP-based schemes is reported in Fig. 5.6. The presented RSLP- and DSLP-based schemes both outperform the benchmark RSTO-based scheme. That is because the RSTO-based scheme just considers selecting a perturbed location outside of the actual semantic location, which neglects the protection of highly sensitive semantic locations. This will cause the overestimation or underestimation of the location privacy. The RSLP- and DSLP-based schemes significantly improve the QoS of the LBS applications, increase the semantic location privacy, and increase the utility of the ship. Moreover, with the continuous perturbation policy space, the DSLP-based scheme can learn a better perturbation policy and further improve the performance of the scheme in comparison with the RSLP-based scheme. The DSLP-based scheme reduces the discretization error compared with the RSLP-based scheme and can avoid converging to the local optimal. For instance, at about the 1500th time slot, the RSLP-based scheme increases the privacy by about 43.43%, reduces the QoS loss by about 2.44%, and improves the utility by 51.16% compared to the RSTO-based scheme. The DSLP-based scheme further improves sensitive semantic location privacy protection performance. For instance, at the 1000th time slot, it improves the privacy by 30.77%, reduces the QoS loss by 15.69%, and increases the utility by 33.33%, compared with that of the RSLP-based scheme.

Figure 5.7 illustrates that the presented sensitive semantic location privacy protection schemes tend to protect the semantic locations with higher sensitivity by inducing the attacker's inference results to less sensitive semantic locations.

Fig. 5.6 Performance of the learning-based semantic location perturbation scheme for a ship in a map with 10 × 10 grids

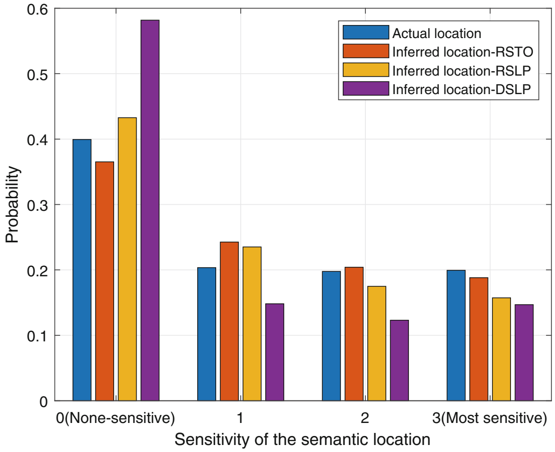


(a) Privacy

(b) QoS loss

(c) Utility of the ship

Fig. 5.7 Comparison of the location sensitivity probability distribution of actual locations and inferred locations with different perturbation schemes

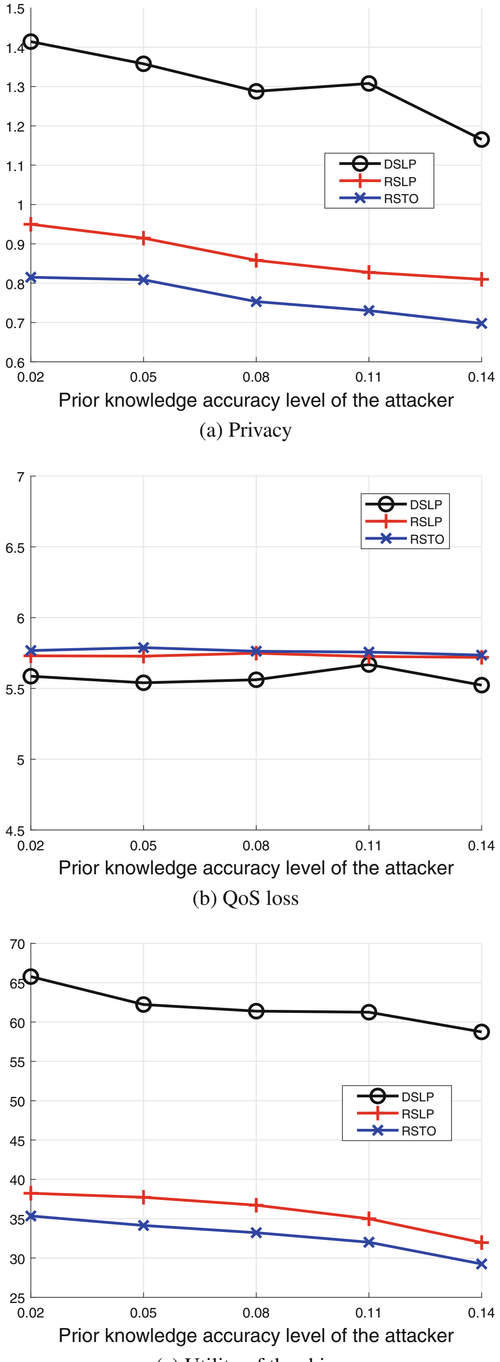


Numbers 0, 1, 2, 3 represent the sensitivity level of semantic locations, in which 3 denotes the location with the highest sensitivity level, and 0 denotes the lowest sensitivity level. For instance, even though the actual semantic location with the lowest sensitivity level 0 is 40.00%, the inferred semantic location with the lowest sensitivity level 0 of the DSLP-based scheme is 59.05%. While the actual semantic location with the highest sensitivity 3 is 20.00%, the inferred semantic location with the highest sensitive 3 of the DSLP-based scheme is only 14.50%, which is 27.50% lower than the actual one. We can see that the inferred probability distribution of locations with different sensitivities of the RSTO-based scheme is close to that of the actual distribution, which means that the RSTO-based scheme fails to protect locations with high sensitivity.

Figure 5.8 illustrates the relationship between the location privacy protection performance and the prior knowledge accuracy level of the attacker. The results show that the average privacy and utility decreases with the prior knowledge accuracy level of the attacker changes from 0.02 to 0.14. That's because the larger prior knowledge of the attacker improves the accuracy of the inference. The prior knowledge accuracy level change has little influence on the QoS loss. For instance, if the prior knowledge accuracy level of the attacker is 0.14 instead of 0.02, the privacy is reduced by 10.85%, and the utility decreases by about 11.86% of the DSLP-based scheme. Note that the DSLP-based scheme can achieve better privacy and utility performance even with a high prior knowledge accuracy level of the attacker. For example, when the prior knowledge accuracy level of the attacker is 0.14, the DSLP-based scheme has 85.51% higher privacy, and 1.02 times higher utility compared with that of the RSTO-based scheme.

Figure 5.9 shows the relationship between the location privacy protection performance and the map size. The results show that the average QoS loss increases with the map size changes from 10*10 to 50*50, and the average privacy and utility of the ship slightly decrease with the map size. That's because the larger map size consists of many more system states, which need more time to derive the optimal

Fig. 5.8 Average performance of the learning-based semantic location perturbation scheme with different prior knowledge accuracy levels of the attacker

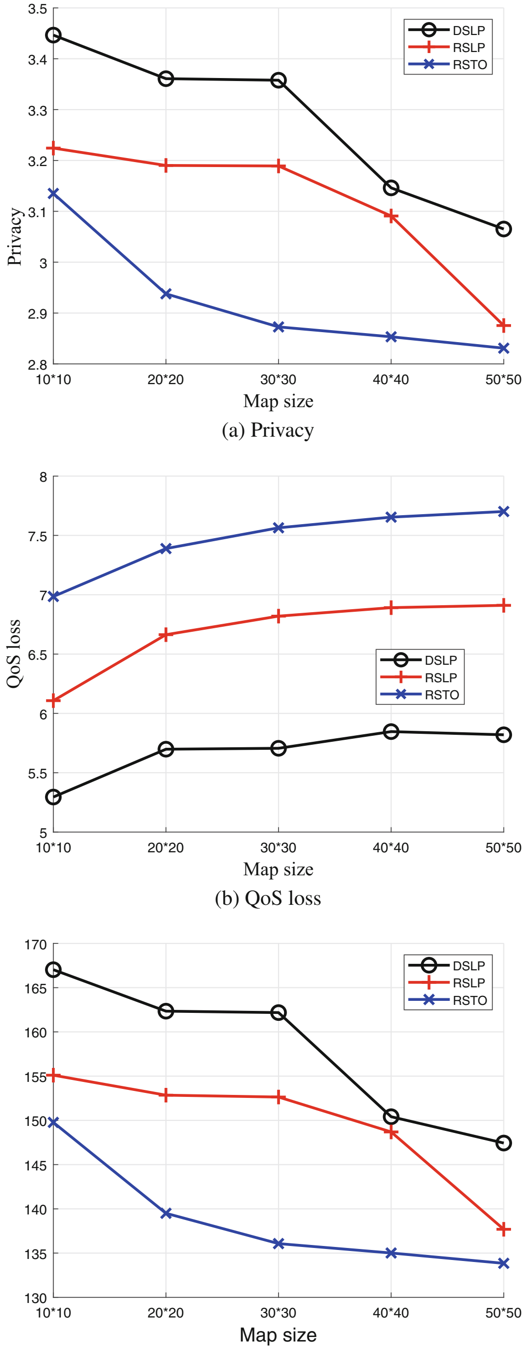


(a) Privacy

(b) QoS loss

(c) Utility of the ship

Fig. 5.9 Average performance of the learning-based semantic location perturbation scheme with different map sizes



(a) Privacy

(b) QoS loss

(c) Utility of the ship

perturbation policy for the ship. For instance, if the ship visits the map size of 50*50 instead of 10*10, the privacy reduces by 6.45%, the QoS loss increases by 56.94%, and the utility decreases by about 12.86%. However, we can also see that the DSLP-based scheme can achieve better privacy and QoS performance even with large map size. The DSLP-based scheme uses DNN as a nonlinear approximator of the Q-value for each perturbation policy to accelerate the learning speed. For example, when the map size is 50*50, the DSLP-based scheme has 9.61% higher privacy, 24.67% lower QoS loss, and 9.33% higher utility compared with that of RSTO-based scheme.

5.6 Conclusion

In this chapter, we have presented an RL-based semantic location perturbation scheme for ships, which protects the sensitive semantic location data while reducing the QoS loss. The ship uses differential privacy technique to perturb the semantic location, and it applies the RL-based scheme to derive the optimal location perturbation policy without knowledge of the interference attack model in the dynamic MNs. A DSLP-based scheme is also presented to select the privacy budget and perturbation angle from a continuous-valued perturbation policy set to further improve the performance of sensitive location privacy protection. Simulation results demonstrate that the presented schemes can increase the privacy, decrease the QoS loss, and thus improve the utility of the ship compared with the benchmark RSTO-based scheme, which does not consider the sensitivity of semantic locations. Although this scheme has been investigated to protect sensitive semantic locations of ships, we believe that the scheme can also be used to protect the semantic trajectory data in location data publishing [42].

References

1. S. Xia, Z. Yao, Y. Li, S. Mao, Online distributed offloading and computing resource management with energy harvesting for heterogeneous mec-enabled IoT. IEEE Trans. Wirel. Commun. **20**(10), 6743–6757 (2021)
2. Y. Zhou, L. Liu, L. Wang, et al., Service-aware 6G: an intelligent and open network based on the convergence of communication, computing and caching. Digit. Commun. Netw. **6**(3), 253–260 (2020)
3. Y. Fu, K.N. Doan, T.Q. Quek, On recommendation-aware content caching for 6g: an artificial intelligence and optimization empowered paradigm. Digit. Commun. Netw. **6**(3), 304–311 (2020)
4. X. Su, L. Meng, J. Huang, Intelligent maritime networking with edge services and computing capability. IEEE Trans. Veh. Technol. **69**(11), 13,606–13,620 (2020)
5. P. Asuquo, H. Cruickshank, J. Morley, et al., Security and privacy in location-based services for vehicular and mobile communications: an overview, challenges and countermeasures. IEEE Internet Things J. **5**(6), 4778–4802 (2018)

6. L. Chen, S. Thombre, K. Järvinen, et al., Robustness, security and privacy in location-based services for future iot: a survey. IEEE Access **5**, 8956–8977 (2017)
7. C.-Y. Chow, M.F. Mokbel, Privacy of spatial trajectories, in *Computing with Spatial Trajectories* (Springer, 2011), pp. 109–141
8. T. Murakami, H. Watanabe, Localization attacks using matrix and tensor factorization. IEEE Trans. Inf. Forensic Secur. **11**(8), 1647–1660 (2016)
9. B. Ağir, Context and semantic aware location privacy, tech. rep., EPFL, 2016
10. C. Lin, D. He, et al., Security and privacy for the internet of drones: challenges and solutions. IEEE Commun. Mag. **56**(1), 64–69 (2018)
11. Z. Tu, K. Zhao, F. Xu, Y. Li, L. Su, D. Jin, Protecting trajectory from semantic attack considering k-anonymity, l-diversity, and t-closeness. IEEE Trans. Netw. Serv. Man. **16**(1), 264–278 (2019)
12. B. Ying, D. Makrakis, H.T. Mouftah, Dynamic mix-zone for location privacy in vehicular networks. IEEE Commun. Lett. **17**(8), 1524–1527 (2013)
13. M. Andrés, N. Bordenabe, K. Chatzikokolakis, C. Palamidessi, Geo-indistinguishability: differential privacy for location-based systems, in *Proceedings of the ACM Conference on Computer and Communications Security*, Berlin, Germany, Nov. 2013
14. B. Ağır, K. Huguenin, U. Hengartner, J.-P. Hubaux, On the privacy implications of location semantics. Proc. Priv. Enhanc. Technol. **2016**(4), 165–183 (2016)
15. I. Bilogrevic, K. Huguenin, S. Mihaila, R. Shokri, J.-P. Hubaux, Predicting users' motivations behind location check-ins and utility implications of privacy protection mechanisms, in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Feb. 2015
16. T. Xia, M.M. Wang, J. Zhang, L. Wang, Maritime internet of things: challenges and solutions. IEEE Wirel. Commun. **27**(2), 188–196 (2020)
17. S.G. Park, S.H. Park, The comparison and analysis of maritime precise positioning using GPS based smartphone. J. Position. Navig. Timing **7**(4), 217–226 (2018)
18. H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, A. Iyengar, Location privacy-preserving mechanisms in location-based services: a comprehensive survey. ACM Comput. Surv. (CSUR) **54**(1), 1–36 (2022)
19. X. Su, S. Jiang, D. Choi, Location privacy protection of maritime mobile terminals. Digit. Commun. Netw. **8**(6), 932–941 (2022)
20. V. Primault, A. Boutet, S.B. Mokhtar, L. Brunie, The long road to computational location privacy: a survey. IEEE Commun. Surv. Tut. **21**(3), 2772–2793 (2018)
21. M. Xue, P. Kalnis, H.K. Pung, Location diversity: enhanced privacy protection in location based services, in *Proceedings of the International Symposium on Location- and Context-Awareness*, Tokyo, Japan, May 2009
22. F. Yucel, E. Bulut, K. Akkaya, Privacy preserving distributed stable matching of electric vehicles and charge suppliers, in *Proceedings of the IEEE Vehicular Technology Conference (VTC-Fall)*, Chicago, IL, Apr. 2019
23. W. Wang, Q. Zhang, Privacy preservation for context sensing on smartphone. IEEE/ACM Trans. Netw. **24**(6), 3235–3247 (2016)
24. C. Dwork, Differential privacy: a survey of results, in *Proceedings of the International Conference on Theory and Applications of Models of Computation*, Xi'an, China, Apr. 2008
25. L. Yu, L. Liu, C. Pu, Dynamic differential location privacy with personalized error bounds, in *Proceedings of the 24th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Apr. 2017
26. R. Shokri, G. Theodorakopoulos, C. Troncoso, et al., Protecting location privacy: optimal strategy against localization attacks, in *Proceedings of the ACM Conference on Computer and Communications Security*, New York, NY, Oct. 2012
27. W. Wang, Q. Zhang, A stochastic game for privacy preserving context sensing on mobile phone, in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Toronto, Canada, Jul. 2014

28. L. Xu, C. Jiang, N. He, Y. Qian, Y. Ren, J. Li, Check in or not? A stochastic game for privacy preserving in point-of-interest recommendation system. IEEE Internet Things J. **5**(5), 4178–4190 (2018)
29. W. Wang, M. Min, L. Xiao, Y. Chen, H. Dai, Protecting semantic trajectory privacy for VANET with reinforcement learning, in *Proceedings of the IEEE International Conference on Communications (ICC)*, Shanghai, China, May 2019
30. R.S. Sutton, A.G. Barto, *Reinforcement Learning: an Introduction* (MIT Press, Cambridge, MA, 2018)
31. M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, W. Zhuang, Learning-based computation offloading for iot devices with energy harvesting. IEEE Trans. Veh. Technol. **68**(2), 1930–1941 (2019)
32. T.P. Lillicrap, J.J. Hunt, A. Pritzel, et al., Continuous control with deep reinforcement learning, in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Juan, PR, May 2016
33. N.D. Lane, S. Bhattacharya, P. Georgiev, et al., An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices, in *Proceedings of the International Workshop on Internet of Things Towards Applications*, pp. 7–12, New York, NY, Nov. 2015
34. S. Chen, J. Hu, Y. Shi, L. Zhao, LTE-V: a TD-LTE-based V2X solution for future vehicular network. IEEE Internet Things J. **3**(6), 997–1005 (2016)
35. S. Chen, J. Hu, Y. Shi, L. Zhao, W. Li, A vision of C-V2X: technologies, field testing, and challenges with chinese development. IEEE Internet Things J. **7**(5), 3872–3881 (2020)
36. M. Götz, S. Nath, J. Gehrke, Maskit: privately releasing user context streams for personalized mobile applications, in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Scottsdale, AZ, May 2012
37. Y. Xiao, L. Xiong, Protecting locations with differential privacy under temporal correlations, in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, Denver, CO, Oct. 2015
38. H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, J. Lu, Fuzzy regression transfer learning in takagi–sugeno fuzzy models. IEEE Trans. Fuzzy Syst. **25**(6), 1795–1807 (2017)
39. K. Chatzikokolakis, C. Palamidessi, M. Stronati, Constructing elastic distinguishability metrics for location privacy. Proc. Priv. Enhanc. Technol. **2015**(2), 156–170 (2015)
40. P. Cheridito, H. Kawaguchi, M. Maejima, Fractional ornstein-uhlenbeck processes. Electron. J. Probab. **8**(3), 1–14 (2003)
41. I. Wagner, D. Eckhoff, Technical privacy metrics: a systematic survey. ACM Comput. Surv. (CSUR) **51**(3), 1–38 (2018)
42. M. Fiore, P. Katsikouli, E. Zavou, et al., Privacy in trajectory micro-data: a survey. Trans. Data Privacy **13**, 91–149 (2020)

Chapter 6
Conclusions and Future Work

In this book, we have discussed maritime communications based on RL to enhance reliability and security performance, including IRS-aided communications, privacy-aware IoT communications, intelligent resource management, and location privacy protection [1–4]. The RL-based IRS communication system chooses the IRS elements and optimizes the signal amplitude and/or phase to enhance the anti-jamming performance with higher received signal strength or improve the secure data rates with cooperative friendly jamming. The maritime communication systems apply RL to optimize the power and spectrum resource allocation to guarantee the QoS in terms of both security and reliability. In addition, learning-based privacy-aware offloading and location privacy protection satisfies the privacy-preserving requirements of maritime ships or sensors.

In this chapter, we summarize the RL-based maritime communication techniques and list the future work in terms of reliability, security, resource management, and privacy protection.

6.1 Conclusions

In Chap. 2, we have investigated IRS-aided secure communication systems to maximize the system secrecy rate of multiple ships against multiple eavesdroppers in time-varying maritime wireless channels and guarantee the QoS requirements. A DRL-based secure beamforming approach jointly optimizes the beamforming matrix at the AP and the reflecting beamforming matrix (reflection phases) at the IRS in maritime wireless networks. Simulation results demonstrate the effectiveness in terms of the secrecy data rate and the QoS satisfaction probability.

In Chap. 3, we have investigated the RL-based task offloading for ship mobile edge computing and the EH IoT devices such as solar energy to extend the battery life, improve the privacy level, and reduce the computation latency against jamming

L. Xiao et al., *Reinforcement Learning for Maritime Communications*, Wireless Networks, https://doi.org/10.1007/978-3-031-32138-2_6

and interference. The RL-based privacy-aware offloading compares the amount of the sensing data, and the offloading data under different channel power states optimizes the offloading policy via trial and error without being aware of the privacy leakage, energy consumption, and edge computation model. The transfer learning and the Dyna architecture can be applied to accelerate the learning process of an IoT device.

In Chap. 4, a distributed cooperative channel assignment and power control approach has been presented to address the massive access management problem in maritime communication networks. The proposed approach supports different QoS requirements (e.g., URLLC and minimum data rate) of a huge number of ships. Furthermore, a multi-agent RL-based scheme was used to realize a centralized training procedure and a distributed cooperative implementation procedure.

In Chap. 5 we have investigated the semantic location privacy protection scheme, where a semantic location's sensitivity is incorporated into the location perturbation scheme to induce the adversary's inference. The proposed scheme releases a less sensitive one instead of a highly sensitive one to improve both privacy and QoS, which can protect both the current semantic location and the potential semantic location in a better way. In addition, an RL-based sensitive semantic location privacy protection scheme is applied to enable a ship user to optimize the perturbation policy in terms of the privacy and the QoS loss. The RL-based scheme can increase both the ship's privacy and the QoS and thus improve the utility of the ship.

6.2 Future Work

In the future, we will investigate how to guarantee stable, reliable, and secure communication services in complex and dynamic maritime environments. In particular, most existing learning-based maritime communication methods suffer from long convergence time due to the random initial exploration, the error and latency to estimate the current network state in each time slot, and the reward estimation difficulty in dynamic maritime systems. New RL algorithms can be investigated to effectively optimize the communication and security policy in highly dynamic communication environments and provide trade-off between exploration and exploitation in the learning process of the maritime IoT devices. For example, some RL algorithms such as policy gradient and double DQN are promising to improve maritime communications [5–7]. Several future research topics and potential solutions for AI-enabled maritime communications are listed as follows.

Computation Efficiency and Accuracy Massive maritime data and complicated networks pose challenges for RL of maritime IoT devices with limited computing resources, e.g., the long learning time to process massive high-dimensional data sometimes exceeds the QoS requirement of the computing intensive maritime services, such as deep learning-based fish tracking. Hence, how to design efficient AI learning schemes to improve both the computation efficiency and accuracy is a significant research challenge. Recent techniques such as residual networks, graphics processing, feature matching, and offline training are promising to improve convergence speed, reduce complex computations, and increase the computing accuracy for maritime applications.

Robustness, Scalability, and Flexibility of Learning Frameworks Maritime networks exhibit high dynamics in the BS associations, channel states, network topologies, and mobility dynamics [8], which bring challenge to the updates of the network in the RL algorithms. Great robustness, scalability, and flexibility of learning frameworks are crucial to support the ever-increasing number of maritime IoT devices and provide high-quality services with various QoS or QoE requirements.

Hardware Development Smart maritime system implementation is challenging, e.g., the RL-based mmWave and THz systems have high-energy consumption and implementation costs, especially for the maritime devices with limited storage and computing energy. Hence, collaboration among hardware components and learning algorithm design is important to effectively handle matrix computations. For example, transfer learning-enabled multi-agent RL algorithm can be designed for smart maritime communications under hardware constraints [9].

Energy Management Smart maritime services in the hot-spot area usually depend on the flexible connection of massive devices with limited battery supply [10]. In the future, we will investigate energy-efficient RL algorithms for maritime IoT devices with harvesting energy (e.g., ambient energy converting and wireless power transfer) to extend the battery lifetime.

In summary, maritime communications involve ship-to-ship communications, ship-to-shore communications, and ship-to-sensor communications [11]. The smart maritime services require reliable, secure, and efficient communications with the guarantee of various QoS requirements.

References

1. S.-W. Jo, W.-S. Shim, LTE-maritime: high-speed maritime wireless communication based on LTE technology. IEEE Access **7**, 53,172–53,181 (2019)
2. C. Zeng, J.-B. Wang, C. Ding, H. Zhang, M. Lin, J. Cheng, Joint optimization of trajectory and communication resource allocation for unmanned surface vehicle enabled maritime wireless networks. IEEE Trans. Comm. **69**(12), 8100–8115 (2021).
3. S. Guan, J. Wang, C. Jiang, R. Duan, Y. Ren, T.Q.S. Quek, Magicnet: the maritime giant cellular network. IEEE Comm. Mag. **59**(3), 117–123 (2021).
4. C. Zhu, W. Zhang, Y.-H. Chiang, N. Ye, L. Du, J. An, Software-defined maritime fog computing: architecture, advantages, and feasibility. IEEE Netw. **36**(2), 26–33 (2022).
5. T. Yang, J. Li, H. Feng, N. Cheng, W. Guan, A novel transmission scheduling based on deep reinforcement learning in software-defined maritime communication networks. IEEE Trans. Cogn. Commun. Netw. **5**(4), 1155–1166 (2019)

6. F. Xu, F. Yang, C. Zhao, S. Wu, Deep reinforcement learning based joint edge resource management in maritime network. China Comm. **17**(5), 211–222 (2020)
7. Y. Liu, J. Yan, X. Zhao, Deep reinforcement learning based latency minimization for mobile edge computing with virtualization in maritime UAV communication network. IEEE Trans. Veh. Technol. **71**(4), 4225–4236 (2022)
8. B. Wang, E. Benli, Y. Motai, L. Dong, W. Xu, Robust detection of infrared maritime targets for autonomous navigation. IEEE Trans. Veh. Technol. **5**(4), 635–648 (2020)
9. Y. Huo, X. Dong, S. Beatty, Cellular communications in ocean waves for maritime Internet of Things. IEEE Internet Things J. **7**(10), 9965–9979 (2020)
10. J. Zeng, J. Sun, B. Wu, X. Su, Mobile edge communications, computing, and caching (MEC3) technology in the maritime communication network. China Comm. **17**(5), 223–234 (2020)
11. Y. Wang, W. Feng, J. Wang, T.Q.S. Quek, Hybrid satellite-UAV-terrestrial networks for 6G ubiquitous coverage: a maritime communications perspective. IEEE J. Sel. Areas Commun. **39**(11), 3475–3490 (2021)