How Reinforcement Learning is Helping to Solve Internet-of-Underwater-Things Problems

Kenechi G. Omeke, Attai I. Abubakar, Lei Zhang, Qammer H. Abbasi and Muhammad Ali Imran
James Watt School of Engineering, University of Glasgow, Glasgow, United Kingdom

***Abstract*—We present a review of how reinforcement learning (RL) is helping to tackle some of the most challenging problems in the Internet-of-underwater-things (IoUTs). Scientists estimate that 50–80% of atmospheric oxygen comes from the ocean, implying that life on earth depends heavily on clean and healthy oceans. This huge significance of the ocean in supporting life on earth is motivating the use of artificial intelligence (AI) and machine learning (ML) tools to create a sustainable marine ecosystem. We briefly review the RL paradigm, its categorisations and RL algorithms developed to solve important problems in IoUTs. New literature keeps emerging that show innovative applications of RL in underwater communications and networking that far outperform conventional solutions and other ML-based methods. Due to its online learning nature, RL is particularly useful for decision making in dynamic environments such as underwater where the communication channel is stochastic and rapidly varying. We explore the applications of RL in IoUTs, showing different classes of IoUTs problems and highlighting RL algorithms that are tailored to solving them. Despite the significant progress that has made in the RL field, there are still many challenges and open research problems in the use of RL in IoUTs. We conclude the paper with an outline of some of these challenges and suggest some ways forward.**

***Index Terms*—internet of underwater things, IoUT, underwater sensor networks, reinforcement learning, machine learning, autonomous underwater vehicles, wireless sensor networks, AUV**

I. Introduction

More than 70% of the earth surface is covered by water, yet less than 10% of the oceans have been explored. Indeed, scientists know more about the deep space than the oceans surrounding our world. The oceans are the primary source of oxygen, energy and food that sustains and supports life on earth, as well as the main source for evacuating excess heat from the earth. Due to the earth's increasing dependence on the ocean, there is need to explore and exploit its resources sustainably through the networking of smart objects in the internet-of-underwater-things (IoUTs).

However, the ocean is also a highly challenging environment for wireless networking. The major technologies used for transmitting information underwater have short range (radio waves and magnetic induction), require line-of-sight alignment between transceivers and are susceptible to fouling (optical communication) or are prone to noise, slow and have low bandwidth (acoustic waves). Acoustic waves are by far the most popular due to their long range. However, the slow speed and the rapidly changing underwater channel lead to Doppler effect complications such as low coherence times and frequency spreading due to motion of underwater sensor nodes, causing severe inter-symbol interferences. It is also plagued by multi-path propagation, absorption, scattering and geometric spreading losses, making acoustic underwater networks bandwidth-, interference- and energy-limited. The highly dynamic nature of the underwater environment implies that heuristics-based and static machine learning methods that map observations to rules or labels often struggle to properly address challenges encountered in IoUTs networking.

Reinforcement learning (RL) uses a reward signal to guide a piece of software (called an agent) to to act in a desired manner in dynamic environments to achieve a desired goal. As a result, it has a rich diversity of applications in the IoUTs and is gaining popularity as a tool of choice for addressing some of the toughest challenges in IoUTs networking. More importantly, learning in most RL methods happens in an online fashion, making it suitable for solving dynamic and sequential decision problems under uncertainty. RL makes underwater networks adaptive, adjusting to changing network parameters in near real time. RL plays a crucial role in keeping underwater networks connected, prolonging network lifetime and maintaining a high quality of service (QoS). It also finds vast applications in manufacturing, gaming, education, marketing, robotics, image processing, healthcare, etc. In this article, we review the most important RL algorithms used in solving IoUTs-related problems.

An overview of IoUTs can be found in [1] while a review of IoUTs was presented in [2], including how ML algorithms are used to enable data analytics at very large scales that cannot be handled by conventional non-ML algorithms. A survey of different applications of deep RL in the IoT industry is provided in [3]. However, the applications captured in [3] do not extend to underwater environments and the peculiar challenges encountered in IoUTs. A summary of recent studies focused on the application of RL techniques in IoUTs is presented in Figure 2. The key applications of different techniques are highlighted while outlining their main contributions and limitations. We note that the bulk of these studies focus on the use of neural networks-based deep learning (supervised ML) techniques in IoUTs. However, surveys/reviews on the use of RL techniques for solving different problems in IoUTs is practically non-existent in the published literature. This article seeks to address this gap by highlighting the advances

This work was funded by the Petroleum Technology Development Fund (PTDF) of the Federal Republic of Nigeria.

The authors are with the James Watt School of Engineering, University of Glasgow, U.K. E-mail: (Kenechi.Omeke, Attai.Abubakar, Lei.Zhang, Qammer.Abbasi, Muhammad.Imran)@glasgow.ac.uk.

in IoUTs applications where RL has been the key driver. For further reading on other applications of RL, the reader is referred to the surveys in the references.

Objectives and contributions: This article provides a review of how RL is used to tackle IoUTs challenges and birth new underwater applications. It highlights state-of-the-art developments in RL that are applicable to IoUTs and provides an overview of different RL algorithms applied in IoUTs, explaining how the algorithms work and the types of problems they address. Finally, the paper reviews current challenges in the implementation of RL in IoUTs and highlights open research problems. A major contribution of this paper is that it connects RL algorithms to problem domains in IoUTs, thereby providing a guide to both new and experienced researchers to understand the relevant RL tools available to them based on the type of problems they are trying to solve. To the best of the authors' knowledge, this is the first review on the applications of RL in IoUTs covering different applications such as routing, energy management, autonomous underwater vehicles (AUV) control and navigation, localisation, etc. The closest other reviews focus only on single applications such as routing [4], AUV navigation and control [5], etc.

II. Overview of Reinforcement Learning

RL involves learning through interaction to achieve some desired behaviour. This usually involves taking trial-and-error actions in an environment, with a numerical reward used to evaluate the desirability of each action taken towards achieving a given objective. If the action taken is the desired one, a high reward is obtained; if the action is an undesired one, a low or negative reward is obtained. An important concept in RL is that time plays a crucial role in RL and feedback is delayed. Current actions can also have long-term impact on the future state (and by extension future rewards) of the system being controlled, so it is sometimes useful to give up actions that yield immediate high rewards for higher future rewards (i.e., to not be *greedy*). Formally, RL can be described using a Markov Decision Process (MDP), which consists of a tuple $M = (S, A, P, R_t, \gamma)$, where S represents the state space, A represents the action space, P represents the environment dynamics and transition probabilities, R_t represents the reward function and γ represents a discount factor that indicates the weights assigned to immediate rewards compared to the one assigned to future rewards.

A. Key RL Concepts

An *agent* is a piece of software or algorithm trained to perform a given task. It works in an *environment*, which is everything external to the agent. A *state* represents all the information used (by the agent) to select the *action* to take at a given time. Each action leads to a *reward*, a numerical feedback signal that indicates the suitability of the agent's actions in the given state, and progresses the agent to the next state. The terms *exploration and exploitation* are often used to describe the trade-off between exploiting current knowledge to take actions that yield high rewards as against trying new actions to potentially discover those with higher long-term rewards.

B. RL Functions

An RL agent can learn three major types of functions, viz:

- **Policy/policy function:** This is a set of rules showing which actions an agent can take in every state. It can be deterministic (every state has an associated action) or stochastic (possible actions in each state is derived from a probability distribution).
- **Value function:** This is a numerical measure of the (long-term) expected total rewards that can be achieved from a given state if a particular policy is followed. Value functions relate to future actions as rewards relate to present actions; a low reward does not necessarily imply a low value function, and vice versa. The value function can either be *state* value function, which depicts how desirable or undesirable a state is, or *state-action* value functions which measures the desirability of a state-action pair.
- **Model (Transition function):** A model is the agent's representation of the environment. It is an ensemble of the environmental knowledge available to the agent which is used to guide its behaviour. A model enables the agent to predict future states of the environment and imagine the outcome of its actions without interacting with the environment.

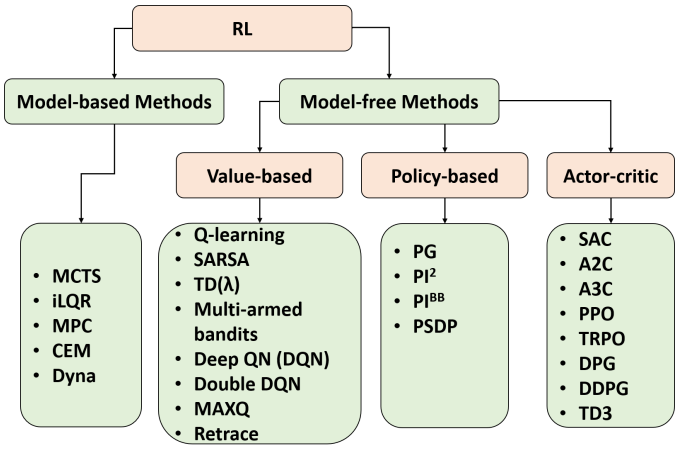

Fig. 1. Broad categorisations of RL algorithms. Model-based methods require a model of the environment for learning whereas model-free methods learn through interaction with the environment. A more in-depth coverage of classifications of RL techniques and the algorithms under each technique can be found in [6].

C. RL Algorithms

RL algorithms can be classified into two broad categories. The first is model-based methods, where the agent directly learns the model or transition function (dynamics) of the environment. The second category is model-free methods, where the agent learns either a policy or value function. We explore each category and their algorithms below.

Model-based methods: These methods combine *planning* and *learning* (ref. [7]). In these methods, an agent either uses a known model of the environment to plan or learns the transition dynamics of the environment (e.g. in a game,

an agent can use a map of the environment to decide the next move). If the agent knows the transition dynamics of the given environment, it can compute an optimal policy without interacting with the environment. These methods require less training time and data samples, thereby improving learning efficiency. However, most real environments are stochastic with unknown transition dynamics, so model-based methods would perform poorly in such environments, especially those with large state and action spaces. Model-based methods include tree-based schemes such as Monte Carlo Tree Search (MCTS) used to solve problems with discrete state spaces and known transition probabilities, analytic gradient computation models such as iterative Linear Quadratic Regulators (iLQR) or Model Predictive Control (MPC), used in learning transition dynamics through repeated interaction with the environment, Dyna, Embed2Control and AlphaZero algorithms.

Model-free methods: Here, the dynamics (transition probability distributions and associated rewards) of the environment is not known to the agent and learning takes place through interaction to directly learn an optimal policy (policy gradient methods) or indirectly learn a value function (value iteration methods), which is used to extract the underlying policy. As a result, algorithms in this category are divided into *value-based* and *policy-based* implementations. Model-free methods are simpler to implement since the agent learns through interaction, but they are not always practical, especially in safety-critical applications. They are particularly suitable for dynamic environments where the optimal action is constantly changing.

Value-based methods: These methods target the optimisation of a state-value function $v^{\pi}(s)$, or state-action value function $Q^{\pi}(s, a)$, from which an optimal policy (argmax of the optimal value function) is obtained. They have high sample efficiency and low variance in estimating value functions but perform poorly in problems with continuous action spaces and suffer from the *curse of dimensionality* problem. Examples of value-based algorithms include Q-learning, state-action-reward-state-action (SARSA), multi-armed bandits (MAB), etc. and Q learning derivatives such as Q-networks (DQN), double DQN, Dueling DQN, Retrace, Noisy DQN, Max Q, etc. Value-based methods can be on-policy or off-policy, depending on whether the policy that is updated during learning or not.

Policy-based methods: Policy-based algorithms optimise a policy directly through policy iteration and hence, are guaranteed to converge to an optimal policy. In essence, these methods directly learn the policy required to take good actions in an environment. They converge faster, require simpler parameters and can be applied to discrete and continuous action problems or both. However, they suffer from high variance and sample inefficiency. Examples of policy-based methods include the REINFORCE, asynchronous advantage actor-critic (A3C), trust region policy optimization (TRPO) and proximal policy optimization (PPO) algorithms. Policy-based RL approaches are covered in [8].

Hybrid methods: These methods learn two or more RL functions such as a combination of model-based and model-free methods, or policy-based and value-based methods. An example is ***actor-critic methods***, which learn a value function and uses it to find an optimal policy. Such methods inherit the advantages (and limitations) of the methods they combine. Examples of actor-critic algorithms include TRPO, PPO, deterministic policy gradient (DPG), deep dDPG (DDPG), twin delayed DDPG (TD3), soft actor–critic (SAC), a derivative of A3C termed advantage actor–critic (A2C), etc [9]. Some algorithms that learn a model of the environment and a value and/or policy function. For example, AlphaGo was developed by DeepMind using MCTS, a value-based method (linear SARSA (λ)), and a policy-based method (REINFORCE) [9].

D. General Applications of RL

RL algorithms are used to solve dynamic problems where hard-coded solutions fail. They are used in decision making, process planning and control operations in gaming, robotics, healthcare, intelligent transport, smart grid, autonomous vehicles, internet of things, etc.

Robotics: RL helps robots develop self-learning capabilities to perform complex operations such as remote surgery, welding, drilling, hold meaningful conversations, navigate crowded spaces, unload heavy goods in confined spaces, etc.

Manufacturing: RL is the brain in many automated machines used to handle large production volumes, improving safety, lowering time and cost.

Advertising: RL can be used to build personalized recommendation systems which learns from the preferences or historical demands of users to improve user experience and lower marketing costs.

Healthcare: RL is used in healthcare for medical diagnosis, drug discovery and development, development of treatment policies, health management, etc. It has already proven successful in the treatment of cancer, HIV, etc., thereby improving health facilities and personnel assignment.

Finance: RL is used in the highly dynamic and unpredictable stock market to build automated trade execution systems by monitoring trends in order to predict future changes in prices, tasks that are nearly impossible for humans.

E. Latest developments in RL

The field of RL is constantly evolving as new research opens up new application prospects. Some of the burning issues at the moment include:

Machine teaching: This involves exploiting domain knowledge of a human expert to design *what to learn* for an RL agent and speed up the learning process, which can potentially enable robots to perform household chores or engage in meaningful conversations [10]. Machine teaching improves sample efficiency and facilitates generalization of machine learning models.

Transfer learning: Involves applying knowledge gained from previous tasks to new related tasks to speed up convergence and achieve optimal or near-optimal performance, which is crucial in safety-critical systems such as self-driving cars and bio-informatics where wrong decisions could have disastrous consequences.

Federated reinforcement learning deals with problems with highly limited data sets and feature state spaces. To

preserve privacy, encrypted output from Q-networks of other agents is shared to enable each agent optimize their individual Q-network, which is useful in manufacturing where various parts of a product are produced by different factories with private decision policies [11].

Explainable AI/RL: This is a move to demystify the decisions and actions of artificially intelligent agents to make them more understandable to their human users, including any expected impact and biases.

III. Applications of RL in IoUTs

RL techniques are effective in dealing with the highly dynamic nature of the underwater environment to overcome the challenges highlighted in Section I. They also provide safety in dangerous and unpredictable aqueous environments where pre-trained supervised learning algorithms would typically fail.

The major application areas are summarised in Fig. 2 and include but are not limited to:

- **Routing:** Multi-hop communication and routing are required in IoUTs applications where direct transmission is not feasible between sensor nodes and the sink. IoUTs networks are resource-constrained; therefore, their routing protocols must possess high resource economy and be able to respond to rapid changes in the environment. Energy efficiency is the primary consideration for long-term applications in the IoUTs. Routing in IoUTs networks is challenging due to frequent link outages, topology changes and partitioning due to node mobility, limited data rates, water motion, turbulence, blockages and the 3D nature of the ocean. The proactive routing protocols used in terrestrial networking have high signaling overheads and are unsuitable for IoUTs networks where frequent topology changes are common. Similarly, geographical-based routing schemes such as vector-based forwarding (VBF), which are popular for underwater networking, struggle when accurate localisation information is not available.

  RL offers advantages of selecting optimal routing policies online, low communication overhead and support for distributed implementation, making it amenable to the dynamic nature of the underwater environment. It has widespread applications underwater, but we illustrate with few examples. To adapt to the constantly-changing link quality, varying traffic and environmental changes, Q learning adaptively selects the best route to the sink, with the link quality worked into the reward function to optimise both energy efficiency and reliability. Making energy efficiency the optimisation goal can address the energy consumption problem in an adaptive manner [12], using single agent or multi-agent RL to achieve network load balancing [9]. Kinematics information or back propagation of Q-values can be used to improve the convergence of such networks and lower transmission failures.
- **Energy management:** Energy efficiency is the most critical consideration for IoUTs networks that are deployed long-term. Energy depletion of nodes leads to network partitioning, causing routing problems and loss of data. Different RL techniques are adopted to optimise energy efficiency in the IoUTs, including energy used by autonomous underwater vehicles (AUVs) in IoUTs networks. For instance, it was shown in [12] that using Q learning to intelligently select cluster heads and implement routing decisions leads to over 42% improvement in energy efficiency compared to popular conventional algorithms. Another Q learning-based algorithm termed QELAR was shown to significantly enhance packet delivery ratio, leading to fewer retransmissions and energy savings [4]. RL is also used for adaptive power management decisions such as duty-cycling, trasmit power allocation, etc. In IoUTs networks with energy harvesting capabilities, RL is employed to deal with the stochastic nature of harvesting routines and traffic demands [13].
- **QoS provisioning:** Some IoUTs applications demand reliable data transfer while others have strict latency or throughput requirements (e.g. military reconnaissance and disaster warning systems). RL is a natural candidate for addressing sequential decision making problems such as QoS provisioning. This is traditionally modelled in service systems using queueing theory, which does not scale well to large networks and perform poorly under realistic network conditions. RL implements an adaptive service-rate controller with probabilistic upper bounds for end-to-end delays in IoUTs systems, thereby providing a guaranteed QoS. The ability to manage and differentiate service requirements while adapting to dynamic changes in the network makes RL particularly appealing for meeting QoS requirements in IoUTs networks.

  Reliability can be achieved on a hop-by-hop (chiefly using ARQ mechanisms) or end-to-end basis (via TCP protocols). ARQ causes poor channel utilisation whereas TCP struggles underwater due to high error rates and long propagation delays. Hybrid techniques such as the segmented data reliable transport (SDRT) combine ARQ and FEC techniques to achieve high reliability, but are still plagued by high overheads. RL responds to network changes in real-time, guaranteeing high reliability irrespective of network conditions. For instance, it can adaptively adjust the data forwarding set of a sensing node according to the network conditions [4]. In good channel conditions, such a neighbour list is shrinked to improve energy consumption and expanded under poor channel conditions to improve packet delivery ratio.
- **Localisation:** For most IoUTs applications, collected data can only be interpreted meaningfully if their locations are known e.g. underwater disaster warning systems. However, GPS is unavailable underwater, so IoUTs nodes use other reference nodes to localise. Traditional localisation estimators such the least squares method (LSM) or convex optimisation methods fail underwater due to the 3D nature of the ocean, node mobility and the physical layer limitations highlighted in Section I.

  A motivating example of how RL addresses underwater localisation challenges can be found in [14]). In addition, Q learning improves the accuracy and energy efficiency of

TABLE I
TABLE OF CHALLENGES, OPEN PROBLEMS AND FUTURE DIRECTIONS

Challenges	Existing Solutions	Future Directions
Lack of suitable prototyping environment	Open AI gym, Stable baselines, Acme	More intuitive environments are desired
Scalability and curse of dimensionality	Deep RL, Federated RL	Hybrid algorithms that combine heuristics with ML
Catastrophic forgetting	Meta learning, Online sample selection, Elastic weight consolidation	Topology-aware weight preservation
Lack of robust algorithms	Multi-agent RL	Stable algorithms that scale with state-action spaces

Domain Challenges	RL Solutions & Example Applications	RL Techniques	Study (Ref)	Year	Pros	Cons
• Frequent link outages • High energy consumption • Rapid network changes	• Adaptive route selection • Network load balancing	Q-learning	[4]	2021	• Fast convergence • High reliability	• Curse of dimensionality • Poor scalability
• Battery-powered nodes • High noise & asymmetric links • Sparse topology	• Load balancing • Energy harvesting • Topology control • Routing	• MAB • Q learning • DQN	[3, 4, 13]	2021	• Low complexity • Adaptive channels	Poor scalability
• Narrow bandwidth • Slow & variable speed • Asymmetric links	• Latency guarantee • Congestion management • Alternate delivery routes	Soft actor-critic networks	[4]	2021	• Fast learning • Highly scalable	• Requires exploration term • Stability issues
• Constant node mobility • High noise levels • High overhead	• Asynchronous localisation • Adaptive localisation • Privacy preservation	Deep Q-networks	[3, 4]	2021	• High dimensionality problems	Lengthy model training
• Localisation problems • Cluttered & unbounded environment • Multi-AUV control	• Asynchronous localisation • Autonomous navigation • AUV Control	• DDPG • Actor-critic networks	[3,4, 6]	2021	• Adaptive control	Unstable if using function approximator
• Protocol selection • Node mobility • Low throughput	• Adaptive transmit powers • Cognitive spectrum • Intelligent time slots	Multi-agent DQN	[10]	2020	• Highly scalable with growing network size	• Non-stationarity problem

Fig. 2. Representative figure of RL applications in IoUTs, showing challenges in different thematic areas, the RL solutions and specific algorithms proposed in the existing literature for addressing the challenges, the strengths and shortcomings of the solutions and relevant references for further reading.

LSM methods by optimising the reference node selection policy without requiring the underlying channel model. Actor-critic networks, Monte Carlo methods, probabilistic collocation, sampling methods remove the need for accurate time clock synchronisation in localisation algorithms by incorporating model uncertainties and external disturbances into the reward function. Function approximation-based dynamic programming solves the local optimum problem in LSM methods by relaxing the linearisation requirement of traditional alternatives to the LSM methods. A privacy-preserving localisation algorithm based on deep RL has also been used to protect reference and target nodes data exchange through information hiding.

- **AUV control, navigation and obstacle avoidance:** AUVs have gained widespread adoption in the IoUTs due to their rapid deployment, reconfigurability, and versatility of applications. They operate in inhospitable environments such as in nuclear storage ponds, deep thermal vents and under polar ice caps and are used in IoUTs for sensing, seabed imaging, mapping, data collection, topology control, etc. However, the endurance and capability of most UAVs are severely limited. RL makes AUVs autonomous and adaptable to complex environments and the open nature of the ocean. It is used for AUV motion planning (path planning/re-planning, obstacle avoidance, cooperative path selection for multiple AUVs, vehicle tracking), docking control, localisation, etc. In the motion planning problem, the AUV interacts with the environment to maximise the reward signal, while respecting the limitations imposed by the physics of the vehicle and avoiding collisions with obstacles and other AUVs. RL solutions are superior to graph search, sampling-based and evolutionary approaches (such as genetic algorithm) in enabling AUVs to independently find optimal paths from one point to another.
RL agents used with AUVs must understand the operating environment and its dynamics. For example, deep RL overcomes the state-space problems of some algorithms and is popular in motion planning applications to reach target points while avoiding obstacles. Model-free RL guides AUVs towards target waypoints by adapting to model uncertainties and external perturbations. Q learning, deep deterministic policy gradient (DDPG) framework (with experience replay to break sample correlations), proximal policy optimisation, actor-critic networks, etc. are some of the common RL algorithms

used in AUV-aided IoUTs [15]. Such RL systems make the AUV adaptive to changing network conditions or external disturbances, input non-linearities and model uncertainties.

- **Resource Allocation:** RL has been employed to achieve optimal power allocation in an IoUTs application by adaptively adjusting the transmit power according to the instantaneous channel conditions, energy and interference levels. It has also been used to manage relay mobility and improve signal transmission under jamming, whereby relay nodes use historical contextual information to select a suitable transmission power that satisfies the required network metrics. RL techniques helps to manage resource allocation in competition-based and handshake-based protocols, as well as MAC control protocols for densely deployed IoUTs nodes. Such protocols must take into account energy consumption (which makes protocols used in delay-tolerant satellite applications unsuitable), long propagation delays, node mobility and poor bandwidth available in underwater acoustic systems. Given these constraints, RL dynamiccally responds to changes in the network conditions and user requirements.

IV. Challenges and Open Research Problems

Some of the challenges facing the practical deployment of RL algorithms in the IoUTs include the exploration-exploitation dilemma, slow convergence of RL algorithms, curse of dimensionality, appropriate design and shaping of the reward function. In IoUTs, one challenge is building a realistic simulation environment and transferring an agent trained in such an environment into the real world to practicalise relevant applications. In addition, there is a dearth of robust algorithms with real-world applicability and adoption. Many current algorithms only work within the framework of simulation environments. Scalability and the curse of dimensionality are also other open problems that extend to nearly all RL algorithms and applications. While deep learning is useful for addressing these class of problems, they require high computational powers and large memories. Some problems with very high-dimensional state and action spaces cannot be handled by current deep learning algorithms. Most past implementations of RL in IoUTs have been limited to problems with low dimensionality. For deep learning-based RL algorithms, scalability is still an issue because to communicate with the agent is realised only through rewards, instead of tuning the network neurons to achieve better predictions, which may sometimes lead to *catastrophic forgetting* as new knowledge erases previous learning. A summary of some of the challenges, current solutions towards addressing them and recommended future efforts are presented in Table I. Addressing these challenges will move RL in IoUTs from the current conceptual stage to practical implementations, thereby creating an explosion in IoUTs and underwater possibilities such as underwater 3D virtual reality, cleaner and smarter oceans, cleaner and efficient maritime and transportation, reliable and rapid early warning systems, safer harbours and oceans, as well as an overall sustainable exploitation of offshore energy resources.

V. Conclusion

This article reviewed areas of IoUTs in which RL algorithms find special applications. Important RL algorithms were presented, along with a discussion of the types of problems they are used to solve and examples of specific applications. A detailed analysis of IoUTs applications that rely on RL techniques was made, with specific algorithms used in different IoUTs contexts highlighted. To the best of the authors' knowledge, this is the first review in which different RL applications in the domain of IoUTs are presented in one paper. Finally, some open research problems in the applications of RL in the IoUTs were outlined and potential solutions explored.

References

[1] M. C. Domingo, "An overview of the internet of underwater things," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1879–1890, 2012.

[2] M. Jahanbakht, W. Xiang, L. Hanzo, and M. Rahimi Azghadi, "Internet of underwater things and big marine data analytics—a comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 23, no. 2, pp. 904–956, 2021.

[3] W. Chen, X. Qiu, T. Cai, H.-N. Dai, Z. Zheng, and Y. Zhang, "Deep reinforcement learning for internet of things: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 23, no. 3, pp. 1659–1692, 2021.

[4] R. T. Rodoshi, Y. Song, and W. Choi, "Reinforcement learning-based routing protocol for underwater wireless sensor networks: A comparative survey," *IEEE Access*, vol. 9, pp. 154 578–154 599, 2021.

[5] L. C. Garaffa, M. Basso, A. A. Konzen, and E. P. de Freitas, "Reinforcement learning for mobile robotics exploration: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[6] H. Zhang and T. Yu, "Taxonomy of reinforcement learning algorithms," in *Deep Reinforcement Learning*. Springer, 2020, pp. 125–133.

[7] T. M. Moerland, J. Broekens, and C. M. Jonker, "Model-based reinforcement learning: A survey," *arXiv preprint arXiv:2006.16712*, 2020.

[8] M. Sewak, "Policy-based reinforcement learning approaches," in *Deep Reinforcement Learning*. Springer, 2019, pp. 127–140.

[9] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE transactions on cybernetics*, vol. 50, no. 9, pp. 3826–3839, 2020.

[10] D. S. Brown and S. Niekum, "Machine teaching for inverse reinforcement learning: Algorithms and applications," in *proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 7749–7758.

[11] H. Hankui Zhuo, W. Feng, Y. Lin, Q. Xu, and Q. Yang, "Federated Deep Reinforcement Learning," *arXiv e-prints*, p. arXiv:1901.08277, Jan. 2019.

[12] K. G. Omeke, M. S. Mollel, M. Ozturk, S. Ansari, L. Zhang, Q. H. Abbasi, and M. A. Imran, "Dekcs: A dynamic clustering protocol to prolong underwater sensor networks," *IEEE Sensors Journal*, vol. 21, no. 7, pp. 9457–9464, 2021.

[13] M. Han, J. Duan, S. Khairy, and L. X. Cai, "Enabling sustainable underwater iot networks with energy harvesting: A decentralized reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9953–9964, 2020.

[14] J. Yan, X. Li, X. Yang, X. Luo, C. Hua, and X. Guan, "Integrated localization and localization for auv with model uncertainties via scalable sampling-based reinforcement learning approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–16, 2021.

[15] Z. Fang, D. Jiang, J. Huang, C. Cheng, Q. Sha, B. He, and G. Li, "Autonomous underwater vehicle formation control and obstacle avoidance using multi-agent generative adversarial imitation learning," *Ocean Engineering*, vol. 262, p. 112182, 2022.