Traffic Prediction and Random Access Control Optimization: Learning and Non-learning based Approaches

Nan Jiang, *Student Member, IEEE*, Yansha Deng, *Member, IEEE*, and Arumugam Nallanathan, *Fellow, IEEE*

***Abstract*—Random Access CHannel (RACH) procedure in modern wireless communications are generally based on the multi-channel slotted-ALOHA (s-ALOHA), which can be optimized by flexibly organizing devices' transmission and retransmission. However, due to the lack of information about the traffic generation statistics and the occurrence of the random collision, optimizing RACH in an exact manner is generally challenging. In this article, we first summarize the general structure of optimization for different RACH schemes, and then review existing RACH optimization methods based on Machine Learning (ML) and non-ML techniques. We demonstrate that the ML-based methods can better optimize RACH schemes compared with non-ML based methods, due to their capability in solving high-complexity long-term optimization problems. To further improve the RACH performance, we introduce a Decoupled Learning Strategy (DLS) for access control optimization, which individually execute two sub-tasks: traffic prediction and access control configuration. In detail, the traffic prediction relies on an online supervised learning method adopting Recurrent Neural Networks (RNNs) that can accurately capture traffic statistics over consecutive frames, while the access control configuration uses either a non-ML based controller or a cooperatively trained Deep Reinforcement Learning (DRL) based controller selected depending on the complexity of different random access schemes. Numerical results show that the DLS optimizer considerably outperforms conventional DRL optimizers in terms of higher training efficiency and better access performance.**

***Index Terms*—Random access, traffic prediction, access control optimization, machine learning.**

I. Introduction

To achieve effective radio access, the random access technique has been integrated into multiple access protocol as a key component of modern wireless communication systems, e.g. Long-Term Evolution (LTE, a.k.a., 4G), Fifth Generation New Radio (5G NR) systems, and etc.. Taking 4G/5G cellular networks as an example, the random access technique is adopted by Random Access CHannel (RACH) procedure, which is used to establish or re-establish connection between unsynchronized devices and their associated Base Station (BS) [1]. The reason to adopt RACH is due to its minimum requirements of priori information, where devices randomly select channels and transmit preambles/packets to the associated BS without negotiation. This uncoordinated transmission inevitably brings uncertainty such that multiple devices may select the same channel at the same time, which results in collided signals that generally cannot be decoded by the BS. Severe collisions occur when massive number of devices simultaneously access the BS, which results in access delay, packet loss, or even service unavailability. In massive Internet of Things (mIoT) scenarios, explosively growing demand for access makes the network overload becoming even heavier, which motivates us to concentrate on RACH in this article.

The RACH framework provides the flexibility of designing access schemes to organize devices' transmission and retransmission. To better manage access, each scheme is aided by several control parameters that are expected to be properly selected at BSs according to communication environments and traffic statistics. However, flexibly selecting these parameters in an exact manner is generally challenging, since BSs rarely know the exact pattern of forthcoming traffic and cannot exactly capture the dynamic of channel. To solve this problem, prior works [2–9] have devoted substantial efforts in designing efficient access control optimization techniques by deriving explicit optimization solutions based on formulated mathematical models. However, the provided access performances are generally limited, due to the high complexity of the problem and the fact that the physics-based model of a RACH system cannot be accurately captured.

In this article, we first briefly introduce the RACH procedure and related RACH schemes in cellular-based networks, then introduce the fundamental mechanism of RACH control as well as classical works in dynamic RACH control, and finally, elaborate that Machine Learning (ML) based access control optimization has potential to better optimize several Key Performance Indicators (KPIs), due to its nature in addressing problems by learning experiential knowledge from the complex environment. Specifically, we introduce the-state-of-art model-free Reinforcement Learning (RL) based RACH control optimization [6, 10, 11], which provides single-step solution to both the traffic prediction and the access control configuration. Knowing that the training solely relies on the interactions with network environment, this approach requires relatively

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC), U.K., under Grant EP/R006466/1. (Corresponding author: Yansha Deng.)

N. Jiang is with the Telecommunications Research Laboratory, Toshiba Research Europe Ltd., Bristol BS1 4ND, UK. This work was done while N. Jiang was working at Queen Mary University of London. (e-mail: nan.jiang@toshiba-bril.com).

Y. Deng is with the Department of Informatics, King's College London, London WC2R 2LS, UK (e-mail: yansha.deng@kcl.ac.uk).

A. Nallanathan is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK (e-mail: a.nallanathan@qmul.ac.uk).

TABLE I: RACH Protocols and Relevant Optimizations

Comparison of RACH Protocols				
Solution	**KPI**	**Parameters**	**Exact Control**	**Reference**
Access Class Barring (ACB)	Success Accesses	Barring factor and barring time	✓	[4, 5, 10]
Dynamic Resource Allocation (DRA)	Time Delay	Channels	✓	[6]
Back-Off (BO)	Success Accesses	Minimum contention window size	✓	[7]
Prioritized Access	Success Accesses	Access Periodicity	×	[8]
Distributed Queuing (DQ)	Success Accesses	Depth and breadth of the tree	×	[9]

Comparison of Optimization Methods					
Type	**Sub-type**	**Complexity**	**Online Adaptation**	**Training Efficiency**	**Reference**
Non-ML based	DA Estimator	Low	×	-	[3]
	MoM Estimator	Low	×	-	[2, 5, 11, 12]
	MLE Estimator	Low	×	-	[4]
ML based	RL-based optimizer	High	✓	Slow	[6, 10, 11]
	SL-based optimizer*	Moderate	✓	Fast	[12]
	DLS-based optimizer	High	✓	Fast	[13]

less domain knowledge of the communication model, however, it also suffers from low training efficiency. To solve these problems, we then introduce a two-step learning framework, namely, Decoupled Learning Strategy (DLS) [13], for access control optimization by decomposing the learning process into two sub-tasks, including traffic prediction and access control configuration. This framework is proposed based on the fact that the forthcoming traffic load is the part of hidden state of the RACH optimization model (to be discussed in Sec. III-B).

The remainder of the article is organized as follows. Section II illustrates the structure and research challenges of RACH. Section III discusses the background of RACH control optimization and reviews classical RACH control optimization methods. Section IV proposes ML-based RACH optimization, including single-step RL-based methods and the DLS framework. Finally, Section V summarizes the conclusion and future work.

II. Research Challenges and RACH Schemes

RACH procedure is responsible for establishing or re-establishing connections between unsynchronized devices and their associated BSs, which is performed by transmitting a preamble from the device along with exchanging of three control signals. The preamble transmission in the first step of RACH can be conducted via two different modes: 1) the contention-free RACH for delayed-constrained access requests (e.g, handover), where the BS distributes one of the reserved dedicated preambles to a known device; 2) the contention-based RACH for delay-tolerant access requests, where a device randomly chooses a preamble from a dedicated preambles pool. In this paper, we solely focus on the latter case. In the following, we first introduce the framework and research challenges of contention-based RACH in the cellular networks, and then describe the classical RACH schemes.

A. RACH framework and research challenges

The *contention-based* RACH relies on the slotted-ALOHA (s-ALOHA) principle, where any device, with a requirement of data transmission, should request access in the first available opportunity. The *contention-based* RACH procedure consists of 4 successive handshaking steps between a device and its associated BS. In step 1, an IoT device transmits a randomly selected preamble (i.e., orthogonal pseudo code, such as Zadoff-Chu sequence) to its associated BS. In step 2, once a preamble is successfully received, the BS replies a Random Access Response (RAR, a.k.a., Msg 2) message via dedicated downlink channel to the device. The BS may not recognize if more than one device transmitting the same preamble, and an RAR message will be replied as usual. In step 3, the device transmits a connection request message (Msg 3) to the BS using the uplink channel scheduled via Msg 2. Those devices received the same identity in Msg 2 will still transmit Msg 3 using the same uplink channel. Consequently, a collision occurs, where the signal of Msg 3 from those devices will not be correctly decoded by the BS. In step 4, the BS replies a contention resolution message with a unique device's identity, and only the device detected its own identity successfully accesses to the network.

A cellular system may be expected to offer connectivity for massive devices. When they access simultaneously via RACH procedure, the network may suffer from a high collision rate. To solve this problem, the BS can organize devices' transmission and re-transmission in a way by using overload control schemes. Recent works [4–9] on s-ALOHA networks have designed effective RACH schemes and optimization techniques to tackle the overload problem. To evaluate the the performance of the novel techniques, a list of KPIs are presented as follows:

- Access success probability: a probability mapping devices to complete RACH within a limited number of attempts.
- Access delay: the time elapsed from the start of RACH to the time receiving the confirmation of success.
- Energy consumption: the total energy consumed during RACH, which is mainly affected by the re-access times.

B. Random Access Schemes

To support massive and diverse access requirements, existing literature have proposed RACH solutions in various wireless

networks, including, but not limited to, LTE, 5G NR, Narrow-Band IoT (NB-IoT), etc.. These solutions are based on the s-ALOHA framework, and share the same purpose of providing more efficient access by alleviating collisions during RACH. In general, the key idea of these solutions is overcoming the channel resource under-provision by intelligently organizing devices' transmission and re-transmission. A classification of existing RACH schemes are summarized in table I and are concluded as follows:

1) Access Class Barring (ACB): devices are forbidden to transmit preamble according to a barring factor within a barring period chosen by BS to alleviate network congestion [4, 5].
2) Dynamic Resource Allocation (DRA): BS allocates a number of channels for RACH according to the requirements during congestion [6].
3) Binary Exponential Back-off (BEB): a device postpones its RACH attempt for a period uniformly selected in random from a range, where its upper bound exponentially increases with the number of RACH failures [7].
4) Prioritized access: devices are splitting into several classes, where the devices from one class are allowed to perform access only in the dedicated access cycle [8].
5) Distributed Queuing (DQ): devices perform access based on a tree splitting algorithm to resolve the collisions by organizing the re-transmission of colliding devices into several distributed queues [9].

III. Conventional Random Access Optimization

Despite that each scheme introduced in Sec. II-B has its own mechanism to control access overload, these schemes are intrinsically based on s-ALOHA protocol, which formulates a general discrete time stochastic control process. In detail, each scheme would divide time into frames, and allows a limited number of devices to execute access using a limited number of channels in each frame. A BS organizes devices' transmission and re-transmission in a centralized manner to facilitate overload control in various traffic scenarios. Taking the ACB scheme as an example, the BS alleviates traffic load by broadcasting a barring probability and a barring period, and some of backlogged devices are forbidden to attempt RACH within a period according to the received barring factors.

A. Research Challenges of Random Access Optimization

RACH optimization targets to identify the optimal strategy of selecting RACH control parameters in real-time to optimize one or more KPIs. This optimal strategy of each RACH scheme is determined by an agent at the BS. More precisely, the agent makes decision at each frame according to observations, which is a set of previous transmission receptions during the RACH, including, but not limited to, the numbers of channels' state in success/collision/idle at the end of each frame. The output of the agent is a set of overload control parameters that will be performed in the forthcoming frame to maximize KPIs. However, obtaining an exact mapping between each observed transmission reception and its optimal RACH configuration strategy is challenging. To tackle these problems, the optimization can be divided into two successive sub-tasks, including (a) traffic load prediction for the forthcoming frame; and (b) RACH control configuration based on the predicted traffic load. Taking the adaptive ACB scheme as an example, by predicting the forthcoming traffic statistic, the number of access success devices can be maximized by choosing the barring factor that yields the forthcoming RACH attempts equal to the number of channels.

1) Traffic Prediction: Traffic prediction can be the most critical problem in s-ALOHA based network, due to the following two challenges: (i) the statistics of forthcoming traffic are generated in a pattern represented by one or mixtures of different traffic types, e.g., periodic, bursty, multimedia streaming, etc., which are hardly captured; (ii) the statistics of accumulated traffic are also hardly captured, since those backlogged devices, either experienced collisions or deferred transmissions based on a RACH scheme's mechanism, are not observable in a BS.

2) RACH Control Configuration: Even with known predicted traffic statistics, maximizing long-term KPIs for RACH in an exact manner is typically challenging, since most KPIs are not only determined by the current configuration, but also correlated with the future configurations. Most non-ML works only optimize the immediate KPI at the next frame, where they ignored the dependency among the RACH control configurations of multiple consecutive frames over the long-term KPI. This simplified assumption of traffic is made due to the limitation in mathematical tool to capture these complex long-term correlation over traffic and the RACH control configurations. For the schemes introduced in Sec. II-B, with the aim of optimizing the number of success access devices, the ACB scheme [5], the resource allocation scheme [6], and the BEB scheme [7] offer exact closed-form solutions, whereas the prioritized access scheme [8], the distributed queuing scheme [9], and the mixture of them only have approximate solutions.

B. Conventional Traffic Estimators

Given a known RACH control configuration strategy, the traffic prediction problem can be cast as a Bayesian probability inference problem, requiring the calculation of the probability for each possible traffic statistics under given previous observation. However, due to the lack of a priori probabilistic model for traffic generation, it is impossible to compute the exact probability of each occurring status. In the following, the existing methods of traffic estimation are concluded as:

1) Drift Analysis (DA) estimator: Given an s-ALOHA system with the stabilized traffic, and a retransmission policy that is geometrically distributed, the evolution of traffic load statistics can be formulated as a Markov chain [3]. The probability that the channel unjams before the backlog increases to a value can be approximately calculated, which yields a maximal backlog value that holds a steady state of the network. However, this fixed step

size scheme is not suitable for networks with unstable traffic, e.g., bursty.

2) Method of Moment (MoM) estimator: MOM has been widely used to estimate traffic load values in s-ALOHA networks [2, 5, 11, 12]. Given a specific traffic load value, the expected numbers of idle, success, and collision channels (i.e., moments) can be easily calculated, and a MoM estimator aims at finding a traffic load value that minimizes the discrepancy between the calculated expectations and its respective observations. For instance, one could calculate the mean absolute error between expectations and observations to yield the MOM-based estimator [12].
3) Maximum Likelihood Estimator (MLE): MLE calculates the maximum likelihood of the optimal Bayes estimator with respect to each traffic load value under each given current observation. This is done in [4] by assuming that, in a frame, devices sequentially and independently select channels one after another, rather than selecting channels simultaneously. This sequential channel selection can be represented by a Markov chain, where the maximum likelihoods for each traffic load statistics of all observations can be calculated using the steady-state probability vector of the Markov chain.

IV. Learning-Based Access Control Optimization

According to the high complexity of access control optimization, ML is a potential tool to provide better optimization performance than conventional methods. Different from the non-ML based algorithms introduced in Sec. III-B, which relies on producing explicit optimization instructions, ML-based access control optimization expects to perform the RACH access control optimization by relying on patterns and inference. These patterns and inference are obtained by training a "machine", also known as a hypothesis class, to discover regularities in data by using computational approach, rather than acquiring domain knowledge via the constructed physics-based model. In the following, we first introduce conventional single-step RL-based access control optimization, and then introduce two-step DLS framework.

A. single-step Reinforcement Learning Based Access Control Optimization

The dynamic optimization of RACH schemes can be formulated as discrete Partially Observed Markov Decision Processes (POMDPs), which is described as a six-tuple {*states, transaction probabilities, actions, observations, observation probabilities, reward*}. Given an agent located at BS aiming at selecting RACH parameters, the agent interacts with the network environment within a sequence of discrete time steps (slots). At each slot, the agent assesses the traffic overload condition of the network relying on environment *states*, and selects an *action* (RACH parameters) on that basis. With a delay of one slot, the network present a new *states* according to the *transaction probabilities*, a *reward* can be obtained by evaluating the selected *action*. The partial observation here refers to that, at each slot, only a limited *observation* can be known by the BS, instead of the exact *states*.

RL technique is a potential solution in addressing discrete POMDPs, due to its capability and scalability in addressing the "curse of dimensionality" in such complex control problems, and its reliance on interacting with the environment without requirements of constructing an accurate environment model [14]. One of the most common applications of RL is in the area of robotics control, which aims at building interactive and goal-seeking RL agents to control robots in complex environments. In general, robot control tasks can be highly time-correlated, where the agent needs to be farsighted to take into account future rewards, e.g., a walking robot maintaining long-term balance requires to consider not only the current status, but also the future statuses. In contrast, dynamic RACH configuration is less complex in the temporal domain due to its target on the relatively immediate reward, while it suffers from much larger action size when multiple RACH schemes being employed at the same time. In these cases, the complexity of dynamic RACH configuration is no less than control problems in robotics. Taking RACH optimization in NB-IoT networks as an example [11], devices are split into three coverage enhancement groups to execute RACH by sharing the same channel, where each group adapts correlated RACH factors including repetition values, preamble numbers, and RACH opportunities. Due to the inter-dependency among all RACH factors, only optimizing each factor in an cooperative manner, instead of independent, can maximize RACH performance, while this results in an discrete action space containing more than fifty thousands possibilities that could much larger than a control problem in robotics. This type of optimization requirement in RACH motivates the utilization of RL.

The RL algorithms have proven to be useful in several applications in the area of RACH control optimization [6, 10, 11, 13]. Recent works [6, 10] have proposed tabular Q-learning methods for the ACB and the resource allocation schemes, which aimed at selecting optimal (*action*) to minimize congestion. Unfortunately, access optimization of one or several other RACH schemes can be more complex than that of the single resource allocation scheme, thus a direct application of the tabular RL algorithms is not feasible due to its low training efficiency. To solve it, Deep Reinforcement Learning (DRL) was adopted to enable learning over a large state space inspired by intelligent game playing [14], while the action space can be broken down into several action variables to be cooperatively trained by multiple agents that solves the oversize action space problem [11]. Moreover, several RACH control parameters are with the continuous action space, e.g., barring factor in the ACB scheme, while straightforwardly adapting Q-learning to solve them by discretizing the action space may degrade access performance. To address this problem, one may use a policy gradient method, e.g., deep deterministic policy gradient, to directly learn policies from the continuous domain of action space [13].

B. Supervised Learning (SL)-Based Traffic Prediction with Conventional RACH Control Configuration

Adopting single-step DRL based access control optimization methods proposed in Section IV-A still face several challenges including: a) the DRL agent is less interpretable and reliable due to the "black box" characteristic; and b) the DRL agent is expected to be updated in an online manner, but the convergence is really slow due to the complexity of the value function as well as the tradeoff between exploration and exploitation. Given a non-ML based RACH control configuration strategy, we can focus on solving the traffic prediction using learning-based method to improve the access performance, namely, SL-based optimizer [12]. This SL-based traffic predictor adopts a modern Recurrent Neural Network (RNN) model, where, different from those conventional methods, the input of the predictor utilizes several previous observations to capture the time-varying trend of traffic for better prediction accuracy.

The RNN predictor is trained by leveraging a novel approximate labeling technique that is inspired by MoM estimators given in Sec. IV. This approximate labeling technique enables online training in the absence of feedback on the exact cardinality of collisions. This online adaptation allows RNN to adapt to the traffic statistics in runtime. In details, RNN is progressively fed with a finite set of observations to produce the forthcoming traffic value for a slot. With one or several slots delay, the traffic value can be estimated by using any heuristic estimator described in Sec. III-B aided by the exact transmission receptions. In this way, the weights of the RNN can be adjusted in order to minimize the error of the former traffic value by prediction with respect to the latter traffic value by estimation.

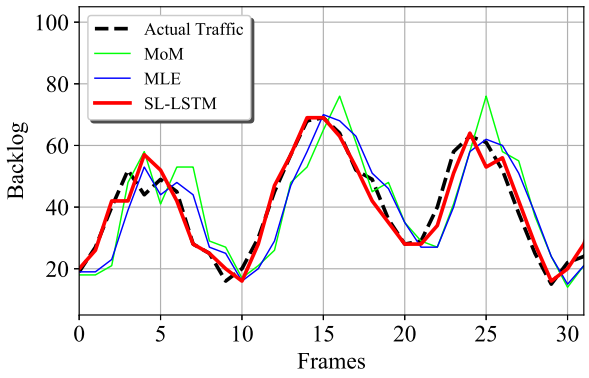

Fig. 1: The actual and predicted backlog of each predictor.

Numerical results are given in Fig. 1 and Fig. 2, which simulated by using Python. In simulations, we set the number of channels as 54, the retransmission constraint as 10, and the traffic as the time limited Beta profile with parameters $(3, 4)$ repeated every 10 frames (The following results in Sec. V.B are also based on these network parameters). Fig. 1 plots the actual and predicted backlog of each predictor, where the SL-based result is employed an RNN with the Long Short-Term Memory (LSTM) architecture. We observe that only SL-

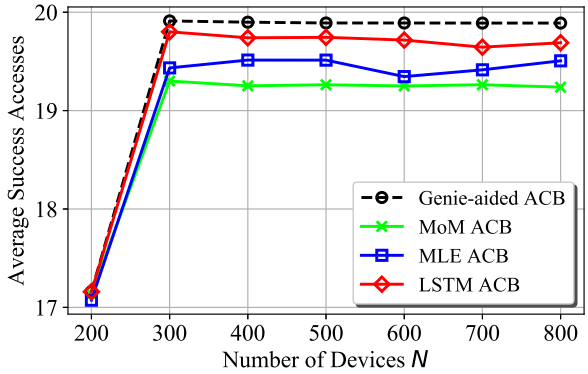

Fig. 2: The average number of success access devices per episode of each optimizer in the ACB scheme.

LSTM can predict the backlog spikes coming from periodic bursty traffic, due to their capability in capturing previous trends of time-varied traffic. Fig. 2 plots the average number of devices which successfully access the network per episode (each containing 100 slots) with the MoM optimizer, the ML optimizer, the SL-based optimizer using LSTM RNN, and the "Genie-aided ACB" (i.e., referring to the ACB scheme aided by actual backlog). Each optimizer solely controls the barring probability, and remains the barring time as 1 slot. It is seen that the SL-based optimizer outperforms the other optimizers, due to its better prediction accuracy. However, it should be emphasized that each optimizer relies on the exact ACB configuration solution. Once the RACH scheme becomes complex (e.g., the hybrid ACB and Back-Off (ACB&BO) scheme and the DQ scheme), the access performance may be degraded due to the ineffectiveness of non-ML based access control configuration.

C. SL-Based Traffic Prediction with RL-Based Access Control Configuration

Given a complex RACH scheme without exact control solution, a DLS has been proposed in [13], which individually executes the RNN traffic prediction and the DRL-based access control configuration as shown in Fig. 3. This method integrates domain knowledge from the communication, that is "the historical and present traffic statistics in the network are directly correlated with the future performance", into learning agents. In details, at each slot, the traffic statistic is first predicted by an RNN predictor as a belief state, and then fed the state into several DRL agents employed in parallel to configure RACH parameters for each RACH scheme. Both the RNN predictor and the DRL agent are updated in an online manner, where the former one relies on the estimated label given in Sec. IV-B, and the latter one uses reward received from observations.

Fig. 4 compares the average number of devices that successfully access the network per episode of the DRL-based optimizer and the DLS-based optimizer for the ACB&BO scheme. It can be seen that the DLS-based optimizer slightly outperforms the DRL-based optimizer due to the fact that

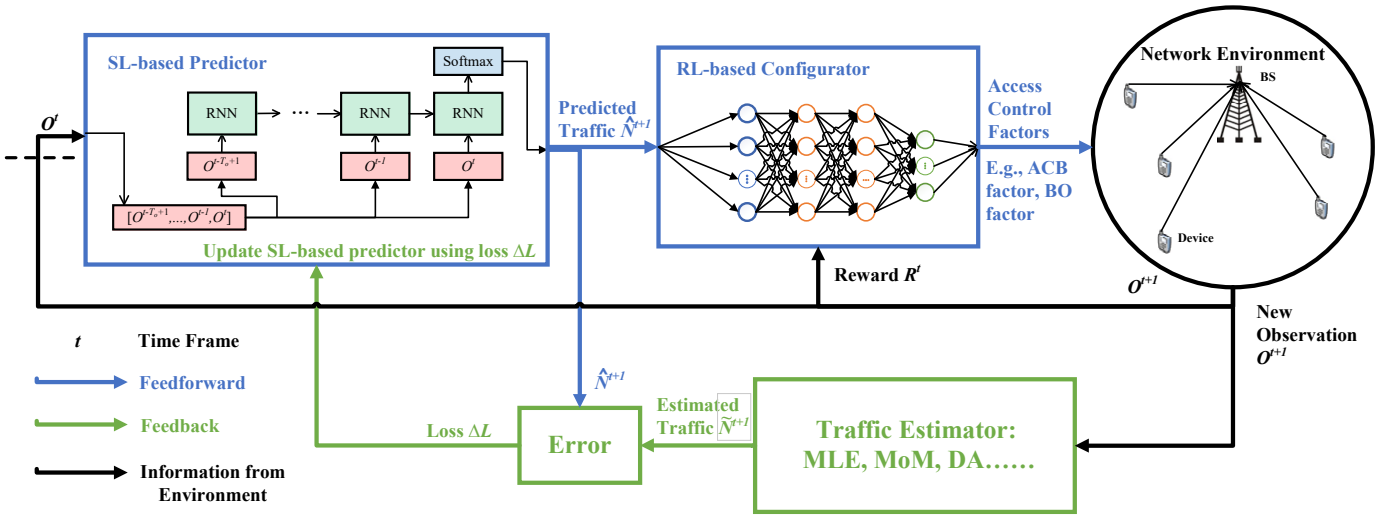

Fig. 3: Illustration of the feedforward and the online adaptation of the multi-step DLS optimizer.

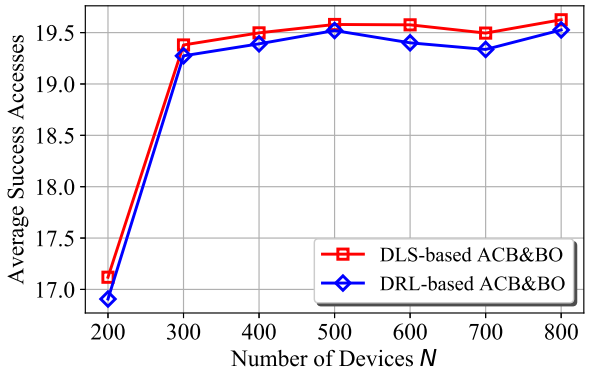

Fig. 4: The average number of success access devices per episode of DRL-based optimizer and DLS-based optimizer in the ACB&BO scheme.

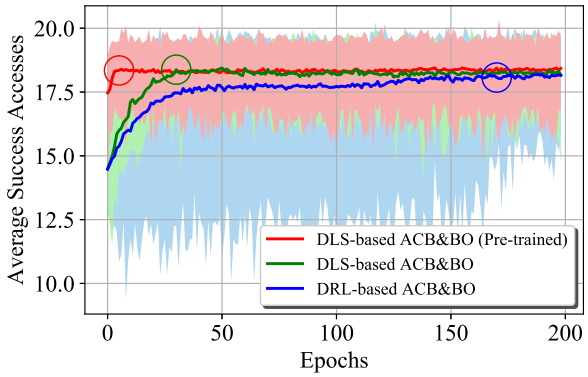

Fig. 5: The required number of episodes for each ML-based optimizer that converges to an efficient solution, where each optimizer has the same hidden layers of neural network and the same hyperparameters for training.

the decoupling simplifies the complexity of the problem. Fig. 5 plots the evolution (averaged over 200 training trails) of the average success accesses per frame as a function in the online phase for "DLS-based ACB&BO (Pre-trained)", "DLS-based ACB&BO (Pre-trained)", and "DRL-based ACB&BO". Here, the "DLS-based ACB&BO (Pre-trained)" refers to that its DRL-agent for access control configuration has been pre-trained, while "DLS-based ACB&BO" is without any pre-training. The approximated converging point of each scheme is highlighted by circles. It is seen that the pre-training can help the DLS-based ACB&BO optimizer to be fairly faster to converge than the one without pre-training. It can also be observed that the training speed of "DLS-based ACB&BO (Pre-trained)" (consumes about 2 episodes) is substantially faster than the DRL-based optimizer (consumes about 170 epochs), which sheds light on its capability of its efficient adaptation.

Due to the milliseconds level requirement of the RACH response time [1], it is highly recommended to employ off-policy learning algorithms in RACH optimization, which decouples feedforwarding and training (all introduced algorithms in this section are off-policy). By doing so, only feedforwarding requires to be processed locally, while the training and updating processes can be deployed at the cloud or an edge by gathering samples from multiple BSs, which would greatly save computational resource. Accordingly, we demonstrate the computational cost of each algorithm in terms of processing time, which simulates on a personal computer with an Intel Core i5-9600K processor. In the simulations, the processing time for one frame of the MOM and the MLE predictor are about 0.007 and 0.043 ms, respectively, whereas that of the LSTM RNN predictor, the DQN optimizer, and the DLS optimizer are about 0.477, 0.677, and 0.734 ms, respectively. Apparently, the ML-based algorithms are more resource-hungry than the non-ML ones. Hence, we evaluate that, even targeting to solve a simple problem, utilizing ML aided by neural networks would still consume much more computational resource than domain-specific solutions, thus we suggest only considering ML techniques for multiple-factors access control problems or single-factor access control

problems without optimal analytical solution. Fortunately, in RACH optimization, the execution time of learning agents is much fewer than 5 ms, which is the minimal period of RACH opportunities in most cellular networks, including LTE, NB-IoT, 5G NR, etc..

V. Conclusion and Future Work

In this article, we elaborated ML techniques to be applied in access control optimization for RACH schemes, which has the potential to play an essential role in realizing efficient access in future wireless networks. The conventional single-step DRL-based optimizer is shown to outperform the non-ML based optimizers in terms of the number of successfully accessed devices, due to that it is capable of learning to master the challenging optimization task. However, the single-step DRL-based optimizer suffers from low training efficiency and the requirement of huge computational resources. To solve this problem, we proposed a two-step DLS-based optimization methods to individually learn the traffic prediction and the RACH control configuration, which considerably improved the training efficiency.

Our results revealed that ML techniques have great potential to revolutionize access control optimization. Compared with the conventional DRL-based method, the proposed DLS-based method can achieve higher training efficiency and better access performance, and can be applied for access optimization of other networks, e.g., grant-free access. Furthermore, we have identified the following future research directions: 1) develop transfer learning and meta-learning for online updating to improve training efficiency; 2) develop distributed learning at devices and BSs to cooperatively guide the transmission decisions; and 3) exploit learning based priority-aware optimization for heterogeneous applications.

References

[1] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-Advanced for mobile broadband*. Academic Press, 2013.

[2] F. Schoute, "Dynamic frame length ALOHA," *IEEE Trans. commun.*, vol. 31, no. 4, pp. 565–568, Apr. 1983.

[3] F. P. Kelly, "Stochastic models of computer communication systems," *J. Royal Statistical Soc.: Series B (Methodological)*, vol. 47, no. 3, pp. 379–395, 1985.

[4] H. He, Q. Du, H. Song, W. Li, Y. Wang, and P. Ren, "Traffic-aware ACB scheme for massive access in Machine-to-Machine networks," in *IEEE Int. Conf. Commun. (ICC)*, May 2015, pp. 617–622.

[5] S. Duan, V. Shah-Mansouri, Z. Wang, and V. W. Wong, "D-ACB: Adaptive congestion control algorithm for bursty M2M traffic in LTE networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9847–9861, Dec. 2016.

[6] S. K. Sharma and X. Wang, "Collaborative distributed Q-learning for RACH congestion minimization in cellular IoT networks," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 600–603, Feb. 2019.

[7] B.-J. Kwak, N.-O. Song, and L. E. Miller, "Performance analysis of exponential backoff," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 343–355, Apr. 2005.

[8] T.-M. Lin, C.-H. Lee, J.-P. Cheng, and W.-T. Chen, "PRADA: Prioritized random access with dynamic access barring for MTC in 3GPP LTE-A networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2467–2472, Jun. 2014.

[9] F. Vázquez-Gallego, C. Kalalas, L. Alonso, and J. Alonso-Zarate, "Contention tree-based access for wireless Machine-to-Machine networks with energy harvesting," *IEEE Trans. Green Commun. Netw.*, vol. 1, no. 2, pp. 223–234, Jun. 2017.

[10] T.-O. Luis, P.-P. Diego, P. Vicent, and M.-B. Jorge, "Reinforcement learning-based ACB in LTE-A networks for handling massive M2M and H2H communications," in *IEEE Int. Commun. Conf. (ICC)*, May. 2018, pp. 1–7.

[11] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement learning for real-time optimization in NB-IoT networks," *IEEE J. Sel. Area Commun.*, vol. 37, no. 6, pp. 1424–1440, Jun. 2019.

[12] N. Jiang, Y. Deng, O. Simeone, and A. Nallanathan, "Online supervised learning for traffic load prediction in framed-ALOHA networks," *IEEE Commun. Lett.*, vol. 23, no. 10, pp. 1778–1782, Jul. 2019.

[13] N. Jiang, Y. Deng, A. Nallanathan, and J. Yuan, "A decoupled learning strategy for massive access optimization in cellular iot networks," *IEEE J. Sel. Areas Commun.*, 2020.

[14] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT, 2018.

Nan Jiang is currently a Research Engineer with the Telecommunications Research Laboratory, Toshiba Research Europe Ltd., Bristol, U.K.. His research interests include internet of things and machine learning.

Yansha Deng is currently a Lecturer (Assistant Professor) with the Department of Informatics, King's College London. Her research interests include internet of things, 5G wireless networks, and molecular communication.

Arumugam Nallanathan is Professor of Wireless Communications and Head of the Communication Systems Research (CSR) group in the School of Electronic Engineering and Computer Science at Queen Mary University of London. His research interests include Beyond 5G Wireless Networks, Internet of Things, and Molecular Communications.