

RESEARCH

Open Access



An architecture and performance evaluation framework for artificial intelligence solutions in beyond 5G radio access networks

Georgios P. Koudouridis¹ , Qing He² and György Dán^{2*}

*Correspondence:
gyuri@kth.se

¹ Wireless Systems Laboratory,
Stockholm Research Centre,
Huawei Technologies Sweden,
Kista, Sweden

² Network and Systems
Engineering, KTH Royal Institute
of Technology, Stockholm,
Sweden

Abstract

The evolution of mobile communications towards beyond 5th-generation (B5G) networks is envisaged to incorporate high levels of network automation. Network automation requires the development of a network architecture that accommodates multiple solutions based on artificial intelligence (AI) and machine learning (ML). Consequently, integrating AI into the 5th-generation (5G) systems such that we could leverage the advantages of ML techniques to optimize and improve the networks is one challenging topic for B5G networks. Based on a review of 5G system architecture, the state-of-the-art candidate AI/ML techniques, and the progress of the state of the art, and the on AI/ML for 5G in standards we define an AI architecture and performance evaluation framework for the deployment of the AI/ML solution in B5G networks. The suggested framework proposes three AI architectures alternatives, a centralized, a completely decentralized and an hybrid AI architecture. More specifically, the framework identifies the logical AI functions, determines their mapping to the B5G radio access network architecture and analyses the associated deployment cost factors in terms of compute, communicate and store costs. The framework is evaluated based on a use case scenario for heterogeneous networks where it is shown that the deployment cost profiling is different for the different AI architecture alternatives, and that this cost should be considered for the deployment and selection of the AI/ML solution.

Keywords: Artificial intelligence, Machine learning, Radio access networks, Network automation

1 Introduction

To accommodate the exponentially increasing traffic demand of Internet users and of connected devices, upcoming 5G wireless networks feature innovative technologies such as infrastructure densification, antenna densification and use of new frequency bands ranging from 700 MHz up to 30 GHz and maybe even higher [1, 2]. Consequently, the wireless networks become increasingly complex, heterogeneous and dynamic [3, 4], which make traditional model-based optimization approaches for radio resource management (RRM) no longer adequate, because highly complex scenarios are unlikely to admit a mathematical description that is at the same time accurate and tractable.

The emerging paradigm of artificial intelligence and machine learning (AI/ML) could provide a promising means for effectively addressing various challenges of legacy model-based optimization approaches. Leveraging recent progress in AI/ML, future radio networks are expected to follow a data-driven paradigm for resource management and for operations, where the level of network automation is increased [5, 6]. In such data-driven network automation paradigm, network nodes are able to determine the best policy based on the experience obtained by processing previous data [7]. The natural question that arises is how to integrate AI-based resource management into the architecture of a radio access network, i.e. where should one store the required data and where should the related computations be executed.

One way of realizing network automation is to have a cloud-based approach in which all intelligence is placed in a central point where data from the network are collected and the computations are executed [8, 9]. Nevertheless, such a centralized approach may have three major issues: latency, privacy and connectivity. These issues may render the centralized architecture a vulnerable solution. As an alternative, in a decentralized architecture, where the data and AI tasks are distributed across the network and the mobile devices, the communication overhead and the traffic load can be significantly reduced. However, due to limited storage and processing capabilities, mobile devices might not be able to develop accurate models. Moreover, the self-organizing nature of the devices may result in poor performance due to a lack of coordination. In between the centralized and completely decentralized architectures there are a variety of hybrid architectures, which—if carefully designed—could combine the benefits of the two. In this article, we propose a framework for comparing three different architectures, in terms of deployment costs and performance metrics, and use the framework comparing the three alternatives for an AI architecture that handles standards-based beyond 5G (B5G) radio access networks.

The rest of the article is organized as follows. In Sect. 2, we present background knowledge about the 5G systems architecture and functions. In Sect. 3, we provide the state-of-the-art ML techniques and architecture solutions as well as the progress of AI/ML in B5G systems. In Sect. 4, we propose the three architectures for the deployment of AI in 5G radio access networks and discuss a framework of requirements and challenges for the evaluation and selection of the architecture solutions. The framework is evaluated, in Sect. 5, based on a multi-hop multi-link use case scenario in heterogeneous networks (HetNets). Finally, we conclude on the article in Sect. 6.

2 5G networks: architecture and functions

One of the fundamental approaches concerning 5G-related 3GPP standards is that the radio access network (RAN) and core network (CN) architectures are described in terms of services and functions. Figure 1 illustrates the 5G general architecture as supported by a set of logical functions divided between the 5G core (5GC) network [10] and the next-generation radio access network (NG-RAN) [11].

The elements of the NG-RAN functions that provide radio access to 5G in new radio (NR) are the access nodes referred to as gNBs [12]. The gNB provides 5G NR access to

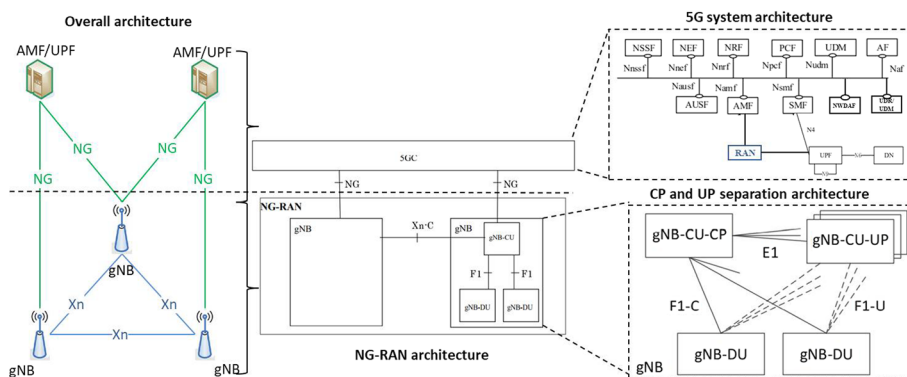


Fig. 1 Generic service-based 5G system and RAN architecture as specified in 3GPP standards

the users by providing NR control plane (CP) and user plane (UP) protocol termination towards the user equipment (UE) side¹. A gNB may be logically split into a gNB-CU (central unit) and one or more gNB-DUs (distribution unit). The gNB-CU is a logical node that hosts the Radio Resource Control (RRC) [13, 14], Service Data Adaptation Protocol (SDAP) [15] and Packet Data Convergence Protocol (PDCP) [16] of the gNB protocols. While the gNB-DU is a logical node that hosts the radio link control (RLC) [17], medium access control (MAC) [18] and physical (PHY) [19] layers of the gNB protocols and functions. A gNB-DU, which is controlled by one or more gNBs, may support a single or multiple cells, while one cell is supported by only one gNB-DU. Furthermore, a gNB may consist of a gNB-CU-CP (central unit control plane), multiple gNB-CU-UPs (central unit user plane) and multiple gNB-DUs. The gNB-CU-CP and gNB-CU-UP are connected to the gNB-DU through the F1-C and the F1-U interfaces, respectively [20]. The gNB-CU-UP is connected to the gNB-CU-CP through the E1 interface [21]. One gNB-DU is connected to only one gNB-CU-CP, and one gNB-CU-UP is connected to only one gNB-CU-CP. The gNB-CU and gNB-DU are interconnected through the F1 interface. The F1 interface supports signalling exchange and data transmission between the endpoints, separates radio network layer and transport network layer and enables the exchange of UE-associated and non-UE-associated signalling [12].

Similar to F1, NG and Xn are logical interfaces. The gNBs are interconnected through the Xn interface [22], while the NG interface [23] allows a NG-RAN consisting of a set of gNBs to be connected to the 5GC (5G core network). The 5GC architecture consists of multiple network functions (NF), including authentication server function (AUSF), access and mobility management function (AMF), data network (DN), unstructured data storage function (UDSF), network repository function (NRF), network exposure function (NEF), network slice selection function (NSSF), policy control function (PCF), session management function (SMF), unified data management (UDM), unified data repository (UDR), user plane function (UPF), application function (AF), security edge protection proxy (SEPP) and network data analytics function (NWDAF) [10]. The 5G system architecture is defined as service-based,

¹ The ng-eNB nodes providing access in E-UTRA and the UE are not depicted in the 5G overall architecture (cf. Fig. 1).

Table 1 The use of spectrum in 5G systems [8]

Band/frequency	Usage
0 MHz band 28	Dynamic spectrum use
1.8 GHz (LTE-FDD)	Software-defined radio (SDR)
2.4 and 5.0 GHz Wi-Fi	Industrial, scientific and medical (ISM) radio band, Wi-Fi, personal area networks, Bluetooth, Zigbee
2.6 GHz (LTE)	Various LTE frequencies incl., LTE Band 3, LTE Band 7 (FDD), LTE Band 20, LTE Band 38 (TDD), and LTE-A
3.5 GHz	5G NR (New Radio) in selected locations
3.5 GHz Band 42	Citizens Broadband Radio Service (CBRS)
5 GHz	Wireless backhaul
Up to 6 GHz	SDR
K-bands (Ku 12–18 GHz), Ka (26–40 GHz)	Satellite bands
6 GHz, 40 GHz, 60 GHz mmWave	In building premises, backhauling, partly SDR, transport network
C-Band, L-Band optical spectrum	Optical transport
Visible light	In building premises

and the interaction between network functions is represented in two ways: (a) a service-based representation, where network functions (e.g. AMF) within the control plane enable other authorized network functions to access their services, and (b) a reference point representation shows the interaction exist between any two network functions (e.g. AMF and SMF).

5G systems and services have been targeted to meet the requirements of a highly mobile and fully connected society. The coexistence of human-centric and machine-type applications in these systems will result in very diverse functional and performance requirements that 5G networks will have to support. In order to meet these requirements in a cost-efficient manner, 5G systems are supposed to leverage a number of technological pillars, such as end-to-end (E2E) network slicing [24], service-based architecture [8], software-defined networking (SDN) [25] and network function virtualization (NFV) [26].

Network heterogeneity in 5G involves the integration of advanced wireless systems, allowing the interconnection of a large variety of end devices. The wireless transport and access network will be based on Sub-6 and mmWave technologies, leveraging massive MIMO with a large number of antennas at the gNBs to improve data rates, reliability as well as energy efficiency [4]. These will coexist with legacy (2-3G), long-term evolution (LTE) and Wi-Fi technologies to allow broad coverage, high availability, higher network density and increased mobility. The use of spectrum in 5G systems is summarized in Table 1.

Advanced 5G use cases and services such as ultra-reliable low-latency communications (URLLC), massive machine-type communications (MMTC) and enhanced mobile broadband (eMBB) place heavy demands on RANs in terms of performance, latency, reliability and efficiency [27]. Meeting these demands may require the adjustment of the RAN's control parameters across time, frequency and space. In general, the potential optimization tasks could be categorized into three domains, as shown in Table 2. The domains are characterized based on the type of parameters involved,

Table 2 Three domains for RAN performance improvement [84]

Domain	Parameter type	Network entities	Update frequency
Network design	Deployment parameters	Basebands, cells, RAN configurations, etc.	Monthly/weekly
Network optimization	Network hyper-parameters	Cell clusters/individual cells	Weekly/daily/ hourly
RAN algorithms	L3 to L1 transmission parameters	Cells and user equipment	Seconds/ milliseconds

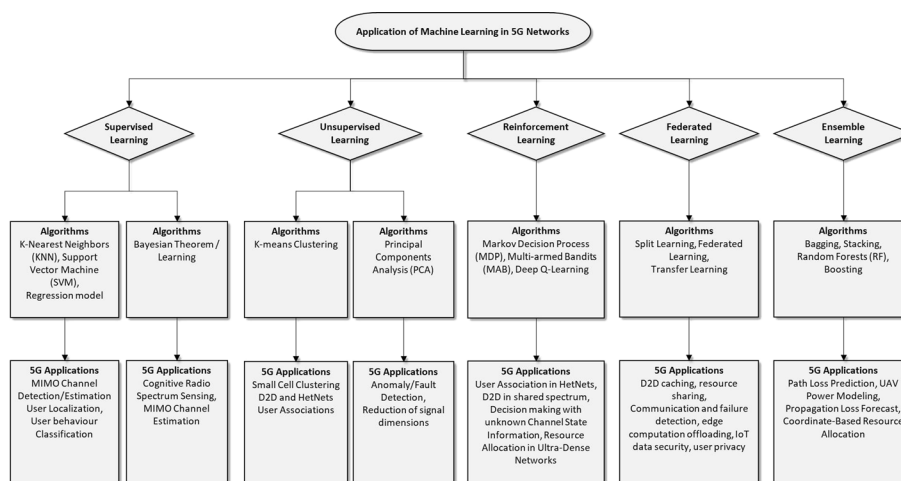


Fig. 2 Machine learning techniques for 5G systems

the type and number of network entities and the frequency at which updates typically take place.

3 AI/ML techniques and solutions in B5G radio access networks

There has been significant interest in using machine learning algorithms for radio network optimization in recent years. The survey [28] presents the application of diverse ML techniques in various key areas of networking across different network technologies. In general, AI and ML can be used to efficiently solve unstructured and seemingly intractable optimization problems in 5G and future communication networks. Advances in the combination of AI/ML and wireless communications reside in different aspects of wireless network design and optimization, including channel measurements, modelling and estimation, physical layer optimization and network management and optimization [73]. In what follows, we provide a structured review of related work on AI/ML for RAN optimization, along the categorization of AI/ML architectures provided in the previous section.

ML techniques and their applications in 5G systems have been discussed in a number of papers, e.g. see the surveys [29, 30] and the references therein. A summary is shown in Fig. 2, which is an extended version of the figure presented in [29]. In what

follows, we give an overview of the most prominent machine learning approaches to network optimization including supervised, unsupervised, reinforcement, federated and ensemble learning.

3.1 Machine learning approaches

3.1.1 Supervised learning

Supervised learning is the task of learning a function that maps an input to an output based on example input–output pairs [31]. The family of supervised learning techniques relies on parameterizable models and labelled data, which allow the estimation of the model parameters. Traditional supervised learning algorithms include regression models, the k nearest neighbour (KNN) algorithm, support vector machines (SVM) and Bayesian learning, while recent interest in supervised learning focuses primarily on deep neural networks.

Regression analysis is based on a statistical process for estimating the relationships among variables. The estimation target is a function (e.g. a linear or logarithm function) of the independent variables. The KNN and SVM algorithms are mainly utilized for classification of points/objects. The above three ML algorithms can be used for estimating or predicting radio parameters that are associated with specific users. For example, in massive MIMO systems associated with hundreds of antennas, both detection and channel estimation lead to high-dimensional search problems, which can be addressed by the above-mentioned learning models. In addition, the KNN and SVM can be applied to finding the optimal handover solutions, which are of importance in a heterogeneous network constituted by diverse cells. At the application layer, these models can also be used for learning the mobile users' specific usage pattern in diverse spatio-temporal and device contexts in [32]. The authors of [32] also show that energy demand prediction is possible with the aid of the centralized KNN algorithms.

The core idea of Bayesian learning is to compute the a posteriori probability distribution of the target variables conditioned on its input signals and on all of the training instances. Some simple examples of generative models that can be learned with Bayesian techniques include the Gaussian mixture model (GM), expectation–maximization (EM) and hidden Markov models (HMM). Bayesian learning models can be applied to spectral characteristic learning and estimation in 5G networks. For instance, the authors of [33] estimated both the channel parameters of the desired links in a target cell and those of the interfering links of the adjacent cells using Bayesian learning with the centralized architecture. Bayesian learning can be utilized in cognitive radio networks. In [34], a cooperative wide band spectrum sensing scheme based on the distributed EM algorithm was proposed, and the authors in [35] constructed a HMM relying on a two-state hidden Markov process. In [36], a tomography model based on the Bayesian inference framework under the centralized architecture was proposed to conceive and statistically characterize a range of techniques that are capable of extracting the prevalent parameters and traffic/interference patterns in cognitive radio networks at both the link layer and network layer.

The above ML algorithms are effective only when sufficient labelled data are available, and they are computationally expensive when training involves large amounts of data. Moreover, it is widely acknowledged that feature engineering could be costly for

the above ML algorithms. These drawbacks motivate the use of deep learning in communication networks. Deep learning refers to multi-layer artificial neural networks, called deep neural networks (DNN), used for tasks such as classification or regression. Compared to the above ML algorithms, deep learning can automatically extract high-level features from data and allows exploiting unlabelled data to learn useful patterns. The state of the art in deep learning techniques with efficient deployment and potential applications to networking was presented in [37]. However, it is worth noting that in general deep learning algorithms are black box models and thus suffer from limited interpretability and explainability. In addition, although deep learning is more capable of handling large amounts of data comparing to traditional ML algorithms, it can be computationally demanding.

3.1.2 Unsupervised learning

Unsupervised learning is a type of machine learning that looks for previously undetected patterns in a data set with no pre-existing labels and with a minimum of human supervision. In 5G systems, the commonly considered unsupervised learning algorithms are K -means clustering, principal component analysis (PCA) and independent component analysis (ICA). K -means clustering aims at partitioning n observations into a number k of clusters. Clustering is a common problem in heterogeneous scenarios associated with diverse cell sizes as well as Wi-Fi and D2D networks. K -means clustering can be utilized for cell clustering in cooperative ultra-dense small-cell networks, for access point association in ubiquitous Wi-Fi networks, for heterogeneous base station clustering in Het-Nets and for load balancing in HetNets [38].

PCA is a technique for multi-variable and mega-variate analysis that can be used to reduce a high-dimensional data set to a lower dimension and reveal some hidden and simplified structure/patterns. The main goal of PCA is to obtain the most important characteristics from data. On the contrary, ICA is a statistical technique that aims to reveal hidden factors underlying random variables, measurements or signals. The goal of ICA is to find new components (new space) that are statistically independent. Both the PCA and ICA constitute powerful statistical signal processing techniques. Their major applications include anomaly detection, fault detection and intrusion detection problems in wireless networks with traffic monitoring. Similar problems may also be solved in sensor networks, mesh networks, and so on. They can also be employed for the physical layer signal dimension reduction of massive MIMO systems, or to classify the primary users' behaviours in cognitive radio networks [39]. In addition, in [40] PCA and ICA are applied to a smart grid scenario to recover the simultaneous wireless transmissions of smart utility meters installed in each home.

3.1.3 Reinforcement learning

Alongside supervised learning and unsupervised learning, reinforcement learning (RL) is one of three basic machine learning paradigms. RL essentially concerns how software agents ought to take actions in an environment in order to maximize the notion of cumulative reward. Specifically, in RL an agent learns to make decisions through trial and error. The considered problem is usually modelled mathematically as a Markov decision process (MDP), where at every time step the environment is in a state, and the agent

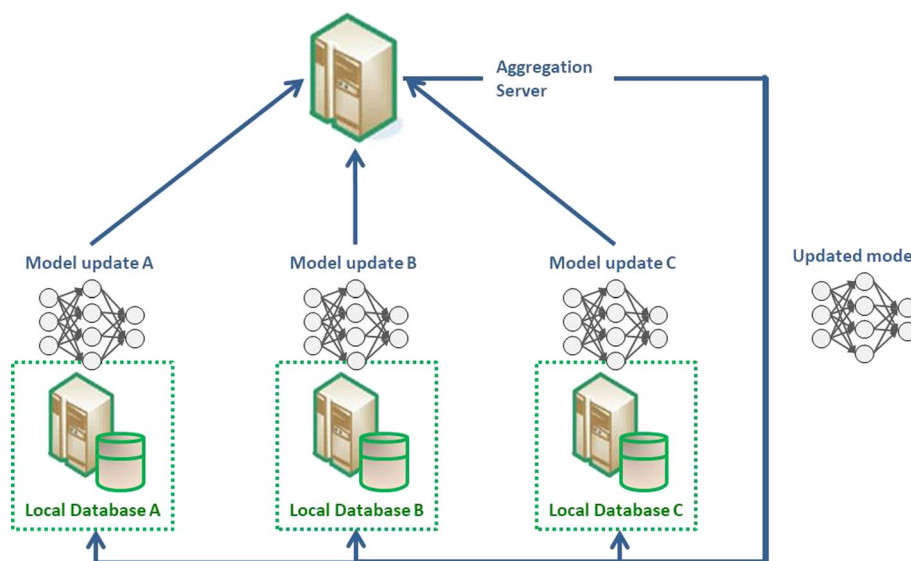


Fig. 3 Federated learning architecture

takes an action and then receives a reward and transitions to the next state according to environment dynamics. During this process, the agent attempts to learn a policy that maximizes its returns (expected sum of rewards). Comparing to other ML techniques, RL is a self-teaching system that does not need labelled input/output pairs nor explicitly presented sub-optimal actions. Hence, it has been efficiently used to enable the network entities to obtain the optimal policy including, e.g. decisions or actions, especially when the state and action spaces are small. However, in many real-life problems like the optimization of the 5G systems, the state and action spaces are very large and thus RL algorithms using the tabular method do not scale. Therefore, in recent studies deep reinforcement learning (DRL), which is a combination of RL with deep learning, has been used to overcome scalability issues. By using DNNs as function approximators, the learning speed and the performance of RL algorithms can be significantly improved.

In [41], the authors study DRL from fundamental concepts to advanced models and provide a literature review on applications of DRL in communication networks. DRL algorithms generally fall into two categories, those based on policy iteration and those based on value iteration. DRL algorithms based on value iteration, e.g. DQN, are more popular in the applications in 5G systems. DRL-based approaches have been proposed to address many emerging issues including dynamic network access, data rate control, wireless caching, data offloading, network security and connectivity preservation which are all important to 5G systems. Moreover, DRL can be applied to efficiently solve classic network optimization problems such as resource allocation, traffic routing and scheduling.

3.1.4 Federated learning

Federated learning is a model training approach that enables devices to collaboratively learn a shared model while keeping all the training data locally [42]. As illustrated in Fig. 3, the general procedure is that devices first download a shared model from a central

server (i.e. aggregation sever), and then, each device trains the model with its locally available data and the changes made to the model are summarized as an update that will be sent back to the server. When the devices send their updated models (e.g. the weights and biases of a deep neural network) to the server, the updated models are averaged to obtain a single combined model. This global model is then sent to all devices. This procedure is repeated for several iterations until a high-quality model is obtained.

3.1.5 Ensemble learning

Ensemble learning is a model training approach that operates on a collection or ensemble of models and combines their predictions by averaging, voting or another combination discipline. As compared to the single modelled ML disciplines described above, ensemble learning approaches can be more expressive and have less bias and variance. The benefit of ensemble learning is that its cost increases only linearly with the number of the models in the ensemble rather than exponentially as it would be the case with a more general model. Assuming that the prediction or classification models are independent—which is a rather strong assumption—ensemble will make more accurate predictions and classifications. There are many ways of creating ensembles including bagging, boosting and random forests. In bagging, a number of models are generated based on an equal number of distinct training sets by sampling with replacement from the original data set. Bagging by means of random forests is a form of decision tree where the ensemble of trees is diverse. In boosting, which is the most popular ensemble method, the bagging is weighted so that correctly classifying or predicting models receive higher weights, while the incorrectly classifying or predicting models receive lower weights. Further improved ensemble learning methods have been developed for various wireless communications problems. In [43], a multiplicity of learning methods including bagging and boosting is compared for the prediction of the path loss, while in [44] ensemble learning is used for propagation loss forecast. Ensemble learning has been also applied to unmanned aerial vehicles (UAVs) for UAV power modelling [45], and sustainable multimodal UAV classification [46]. Further applications include the exploitation of a deep ensemble-based wireless receiver architecture for mitigating adversarial attacks in automatic modulation classification [47], and coordinates-based resource allocation based on random forests [48, 49].

3.2 AI/ML for 5G network management and optimization

State-of-the-art work on network management and optimization using ML technologies in wireless is rather extensive. In this section, we provide a list of the most relevant research studies in tabular format with some remarks and a classification of the suggested solutions. Observing the research work listed in Table 3, one can see that ML-based approaches could effectively solve non-convex and complex problems in various network contexts, e.g. joint user association and transmission scheduling, to achieve various goals including throughput maximization and energy consumption minimization. At the same time, these problems are hard to be addressed by traditional model-based optimization techniques. Moreover, with decentralized DRL-based approaches, network entities can make observation and obtain the best policy locally with minimum

Table 3 Network management and optimization for 5G networks using ML

Use case	Objective and scenario	Approach	AI architecture	Refs.
User association	Delay minimization under energy consumption constraint in ultra-dense network with edge computing	DRL	Decentralized	[50]
User association	Load balancing with service rate selection in vehicular network with heterogeneous base stations	DRL	Centralized	[51]
Cell selection	Capacity maximization and number of handovers balancing in open access femtocell network	Q-learning	Decentralized	[52]
User association	UE outages minimization by cell range expansion for Picocell with bias value	Q-learning	Decentralized	[53]
Network slicing	Maximize network revenue by intelligently admitting network slice requests	RL	Centralized	[54]
Network slicing	Maximize the utility of individual service provider by joint slicing computing and communication resources	DQL	Decentralized	[55]
Traffic prediction	Short-term traffic prediction for individual users	Meta-learning	Decentralized	[57]
Resource allocation	Minimize service latency in a sliced RAN by computing resource allocation and task transmission scheduling	DRL	Centralized	[56]
Traffic prediction	Predict the maximum service-specific traffic load for each slice based on real-world 5G network data	DL	Centralized	[58]
Content placement	Track and predict time-variant content requests from users	DL	Centralized	[59]
Content delivery	Determine caching and computing offloading decisions to reduce operational cost on edge server	DRL	Centralized	[60]
Rate adaptation	Video quality-aware rate control for real-time video streaming	DRL	Decentralized	[61]
User association	Coverage and capacity maximization for massive MIMO systems	DRL	Centralized	[62]
User scheduling	Sum rate maximization with joint user-cell association and selection of number of beams in 5G mmWave networks	Transfer Q-learning; Unsupervised learning	Hybrid	[63]
User scheduling	Power-efficient resource allocation in cloud RANs	DRL	Centralized	[64]
User scheduling	Minimizing the age of correlated information in IoT systems	DRL	Centralized	[65]

information exchange among each other. This will not only reduce communication overhead but also improve security and robustness of the networks. In addition, DRL could be used as an efficient tool to solve certain problems that can be modelled as a non-cooperative game, such as, cyber-physical attacks, interference management and data offloading.

3.3 AI/ML for 5G in standards

Recently, there have been initiatives to apply AI/ML to 5G networks and beyond in standards organizations including the International Telecommunication Union (ITU) and Third Generation Partnership Project (3GPP), as well as other study groups such as FuTURE, Telecom infra project (TIP) and 5G PPP, as shown in Table 4.

Table 4 AI/ML for 5G networks and beyond in standards [73]

Leading organization	Group name	Starting time	Purpose	Applications
3GPP	NWDAF	March 2018	Allow 5G operators to monitor the status of a network slice or third-party application performance	5G core network data analytics
5G PPP	CogNet	July 2015	Build an intelligent system of insights and action for 5G network management	Autonomic network management based on machine learning
FuTURE	Wireless big data for smart 5G	November 2017	A white paper that collects a pioneering research works on big data for 5G in China	Boost spectrum efficiency and energy efficiency, improve the user experience and reduce the cost
ITU	ML5G	November 2017	Identify relevant gaps and issues in standardization activities related to ML for future networks	Interfaces, network architectures, protocols, algorithms, and data formats
TIP	AI and applied ML	November 2017	Define and share reusable approaches, proven practices, models and technical requirements for applying AI and ML	ML-based network operations, optimization and planning; customer behaviour-driven service optimization; and multi-vendor ML-AI data exchange formats

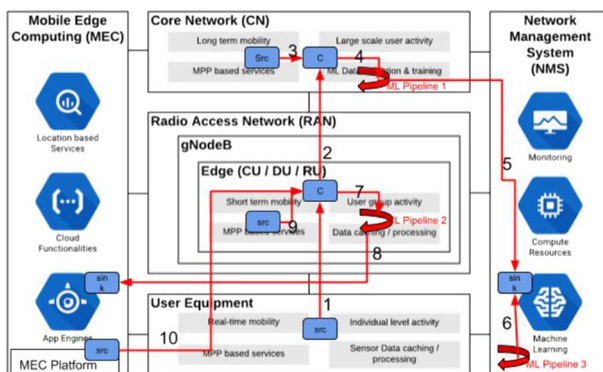


Fig. 4 Hosting of multi-level ML pipeline in 3GPP, MEC [5]

In Nov. 2017, ITU started a focus group on “Machine learning for future networks including 5G (FS-ML5G)” at its meeting in Geneva. The focus group is responsible to draft technical reports and specifications for ML for future networks, including interfaces, network architectures, protocols, algorithms and data formats [5]. The FS-ML5G was active from January 2018 until July 2020. During its lifetime, FG-ML5G delivered ten technical specifications, covering the study of architecture, interfaces, use cases, protocols, algorithms, data formats, interoperability, performance, evaluation and security. Among these specifications, they propose a unified logical architecture for ML in 5G and future networks [66] and give an instance of realization of the

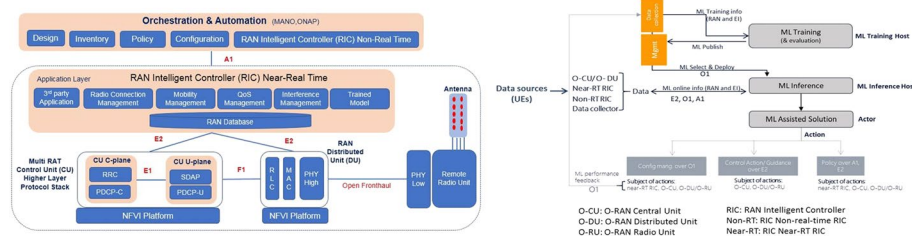


Fig. 5 ML training host and inference locations in O-RAN [6]

logical architecture on a 3GPP system along with MEC and management systems, as shown in Fig. 4.

The FG-ML5G has provided more than 30 use cases and their requirements. Notable among them are RRM for network slicing, end-to-end network service design automation and end-to-end fault detection and recovery. To address RRM problems with reduced complexity and cope with the growing variety of scenarios, in [67], the authors propose a lean RRM architecture that consists of one or a few learner(s) that learn RRM policies directly from the data gathered in the network using a single general-purpose learning framework, and a set of distributed actors, which execute RRM policies issued by the learner and repeatedly generate samples of experience. In [68], the authors adopt ML to realize cognitive network management in support of autonomous networking. They have employed ML to minimize the role of humans in the control loop and present a use case of cognitive security manager for anomaly inference and mitigation over a software-defined infrastructure.

In 2018, the 3GPP standards group proposed the network data analytics function (NWDAF), a collection of interfaces, so as to allow the definition of analytics functions that can be applied to monitoring the status of a network slice or the performance of a third-party application on the “Zero Touch and Carrier Automation Congress” [9]. The NWDAF forms a part of 3GPP’s 5G standardization efforts and could become a central point for analytics in the 5G core network. The NWDAF is still in the early stages of standardization, but could become an interesting enabler for innovation.

Complementary to the 3GPP standardization targets, the O-RAN initiative defined the O-RAN architecture which is empowered by the principle of openness and is expected to have a major influence on the next-generation networks. In Fig. 5, we show the reference O-RAN architecture, and the general framework for AI/ML functions and interfaces in O-RAN [6].

The white paper [7] published on the FuTURE Forum is a collection of pioneering research works on big data for 5G in China, in both academia and industry. It proposed the concept of “smart 5G” and argued that the 5G network needs to embrace new and cutting-edge technologies such as wireless big data and AI to efficiently boost both spectrum efficiency and energy efficiency, improve user experience and reduce cost.

TIP launched a project group, “AI and applied machine learning” in November 2017 [69]. The group applies AI and ML to network planning, operations and customer

behaviour identification to optimize service experience and increase automation [70]. The objective is to define and share reusable, proven practices, models and technical requirements for applying AI and ML to reduce the cost of planning and operating telecommunications networks, understand and leverage customer behaviour and optimize service quality for an improved experience.

5GPPP has also launched its efforts on combining AI with wireless communications which aims to build an intelligent system of insights and action for 5G network management [71]. These developments in standards and study groups aim to use AI for physical layer and network management, which will greatly boost the performance of wireless networks.

4 AI/ML architecture performance framework for RAN

Employing AI and ML techniques is of importance for the advancement of wireless communication networks. In [72], the authors investigate network design and operation using data-driven approaches, compare them to traditional model-based design techniques and conclude that we are rapidly reaching the point where the quality and heterogeneity of the services we demand of communication systems will exceed the capabilities and applicability of present modelling and design approaches. To this end, an automation of the modelling and design processes is to be spawned which imposes a new set of requirements on the network functions that generate and execute the models both in terms of their architectural approach and their performance gains. The architecture approach deals with the deployment of the functions as supported by the AI/ML learning and inference approach, while the performance deals with the efficiency of the approach to solve a particular use case at the cost associated with it. To address this issue, we define an AI/ML architecture performance framework for RAN, as detailed further in the remainder of this section.

4.1 RAN architecture aspects

Following the common terminology and definitions in the O-RAN architecture [6], an ML workflow is a process consisting of data collection and preparation, model building, model training, model deployment, model execution, model validation, continuous model self-monitoring and self-learning/retraining related to ML-assisted solutions. We refer to the network function that hosts the training of the model as the ML training host, and to the network function that hosts the ML model during inference mode (i.e. model execution and model update if possible) as the ML inference host. When designing ML-enabled RANs, depending on the locations of data sources, the ML training host, the ML inference host and the point of actuation, three types of AI architectures can be identified: centralized, completely decentralized and hybrid.

In Fig. 6, we show a simple illustration for the three AI architectures. The nodes in the RAN are categorized in three kinds:

- a central *controller or coordination point* (CP), which is equipped with a centralized processing unit and a data storage unit, operating above or in the CU-CP or CU and operating over multiple CU-UPs or DUs, respectively

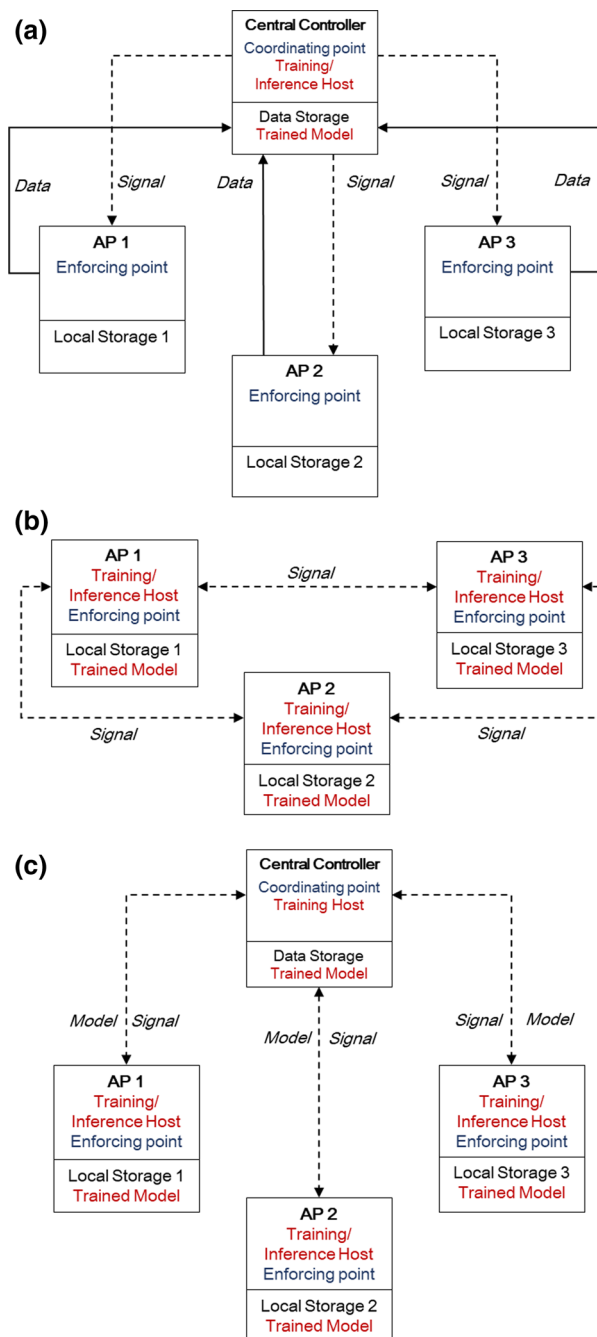


Fig. 6 Illustration for AI architecture alternatives. **a** Centralized, **b** decentralized and **c** hybrid

- an *access point* (AP), which is typically equipped with micro processing unit, e.g. BBU and a local database, operating in gNB nodes at CU-UP or DU and
- a *terminal point* (TP), which corresponds to UEs and mobile devices with limited processing capacity and storage units.

In a network with centralized AI architecture, the training host is located at the CP, where the information needed for model training is collected from all APs. The

centralized approach could take advantage of the computational capability and the data storage of a data centre, and hence, it may facilitate training complex neural networks and coordinate access points (AP) and UEs in the network. This training coordination is achieved at the price of a large amount of data transmission and control signalling overhead. The data transmission is expected to be excessive depending on the number of parameter and parameter values to be exchanged and the complexity of the AI/ML model to be built. When a model is trained and built by the CP, it is then transferred to the AP for decision and optimization inference.

In the presence of capacity constraints on the fronthaul/backhaul links and potential privacy/security issues, it is expected that various network functions could be executed locally at APs or with minimal information exchange in transport networks. Doing so results in a decentralized architecture. Compared to a centralized architecture, the major drawback of the decentralized architecture is that locally trained models cannot make use of information from remote network entities and thus a decentralized architecture may lead to inferior performance in lack of global coordination. In this framework, a decentralized architecture is manifested in its extreme where control decisions are derived locally and are supported only by local data with no data exchange or coordination among the APs. Any form of coordination among the APs constitutes a form of hybrid architecture, as in, for example, federated learning.

Hybrid architectures may combine the best of centralized and decentralized architectures. Typically, ML algorithms under the hybrid architecture are trained locally by their respective APs, and the central server in the CP is used as a coordinator to orchestrate the different steps of the algorithms and coordinate all the participating nodes during the learning process. In general, local data samples in APs will not be collected by the CP; instead, the coordination should be done by exchanging parameters (e.g. the weights and biases of a deep neural network) between these local APs and the CP. Apart from signalling parameters and depending on the learning technique, the communication between the AP and the CP may include the exchange of models in both directions. An example hybrid learning scheme is federated learning [42], as illustrated in Fig. 3.

A particular network may employ multiple AI architectures for different optimization tasks. In [73], the authors show deep learning models that can be applied to cloud, fog and edge computing networks, where the cloud network is the data and computing centre, the fog network includes many nodes, and the edge network contains many end users and devices. Considering the available communication bandwidth, besides ML models executing at the cloud, decentralized learning, classification and signal processing algorithms are needed, using lightweight learning models (i.e. ML models running with limited storage capabilities, computational power and energy resources). As doing so, the advantages of decentralized and centralized algorithms could be combined, thereby trading off complexity, latency and reliability. In [74], the authors propose a multi-layered control architecture for deploying and implementing various ML applications in cellular networks with edge computing. In this architecture, each RAN controller (CU and DU) is associated with a cluster of gNBs and is deployed in a mobile edge cloud (MEC), so as to minimize the communication latency. The RAN controllers are responsible not only for RAN control, but also for running the data collection and ML

Table 5 List of KPIs from 5G RAN functions [76]

Level	RAN function	KPI
UE-level	Radio channel info at DU	Channel quality indicator (CQI)
UE-level	Radio channel info at CU-CP for serving/neighbouring cell	RSRP, RSRQ, SINR [13, 14]
UE-level	L2 info at gNB	Average/distribution DL UE throughput, packet delay and RAN part packet delay components, data volume (per QCI, [77]), DL PDCP occupied/unused buffer size, packet loss rate per DRB/logical channel [78]
Cell-level	L2 info at DU	MCS distribution in PDSCH, total error number of DL/UL TBs (per layer at MU-MIMO case [79, 80]), average/distribution DL UE throughput in gNB (per QoS, 5QI or CQI level and per slice [79]), RAN part packet delay components [78], packet delay (per QCI [77])
Cell-level	L2 info at CU-UP and DU	Packet delay (per QoS, 5QI or CQI level and per slice [79])
Cell-level	L2 info at CU-UP or DU	Packet loss rate (per QoS/5QI/QCI level and per slice [79, 80]), DL packet drop rate (per QoS/5QI/QCI level and per slice [79, 80])

infrastructure. The proposed architecture could enable the RAN controllers to implement machine learning techniques at the edge of the network.

4.2 RAN performance aspects

The performance of RAN can be evaluated differently depending on the evaluation objectives and its operation scenarios and use cases. Different use cases consider different scenarios, such as macro-cell deployments, ultra-dense networks and/or multi-hop wireless HetNet consisting of micro-cells, integrated access backhaul (IAB) nodes, indoor and outdoor UEs [75]. In RAN, a typical algorithm runs in CU or DU to solve a RRM-related optimization problem, such as user association, user scheduling, cell handover, load balancing, interference coordination, coverage and capacity provision and admission control, which are some of the fundamental problems in RRM. These problems define their own objectives, and many of them are combinatorial optimization problems that are in general hard to solve using traditional approaches.

Therefore, the use case-related set of performance metrics are crucial to be employed to evaluate AI/ML-based approaches with different architectures. To quantitatively evaluate the proposed AI/ML solutions under different AI architectures, apart from the use case-related network performance metrics, a second set of metrics need to be considered. This second set of metrics involves the AI architecture alternatives and is defined by the cost to achieve the network performance. Combining these two sets of metrics, a performance cost metric can be defined to comparatively evaluate AI/ML architecture solutions.

4.2.1 Network performance

The network performance refers, in general, to the effect of the learning of a context and, in particular, to the set of network performance aspects, e.g. capacity, reliability, latency, etc., and the metrics to measure them. Consequently, it is use case specific and the evaluation is performed based on the key performance indicator parameter values of the use case as formulated by its objective function. To evaluate the network

performance, a natural metric is the achieved objective value, e.g. the user throughput, the packet delay, packet loss, etc. Table 5 provides a list of primary metrics and key performance indicators (KPIs) that can be provided at different levels and functions of the NG-RAN architecture. Besides the primary metrics that are defined in the 3GPP standards, further metrics such as the age of information, the fairness and the age evolution for each UE can be derived and compared to the classic network performance metrics such as throughput and delay. As the RAN systems increase in complexity, different network performance metrics are expected to arise for differentiating the network performance.

4.2.2 Learning performance

The second set of metrics concerns the requirements and the overhead of the learning system. For this discussion, a logical architecture of the components of the learning system is first introduced followed by a definition of the metrics.

The components of the logical architecture are as follows:

- *Training node* The entity in which data are used for training a ML model. In a centralized architecture, there is a single training node, while in a decentralized or hybrid architecture there are multiple training nodes. Multiple training nodes could be co-located on the same physical resource. We denote by \mathcal{T} the set of training nodes.
- *Inference node* The entity in which data are used for make inference using a ML model. In a centralized architecture, there is one or multiple inference nodes, while in a decentralized or hybrid architecture there are multiple inference nodes. Multiple inference nodes could be co-located on the same physical resource. We denote by \mathcal{I} the set of inference nodes.
- *Enforcement point* The location where decisions are implemented based on output from the inference nodes, e.g. a baseband unit (BBU). We denote by \mathcal{E} the set of enforcement nodes.
- *Data source* The locations where data required for training and/or inference originates from. For training, the data need to be delivered to the training nodes, either in real time or using bulk transfer, depending on the learning model. For inference, data need to be delivered to the inference nodes, likely in real time. We denote by \mathcal{S} the set of data sources.

In Fig. 7, we illustrate the logical nodes under the three AI architectures. Note that the illustrations do not reflect the physical location of each node. In a network with centralized AI architecture, the training node collects data from data sources across the network and trains the model accordingly. The inference node uses the trained model to make decisions and send the decisions to the corresponding enforcement nodes, which execute the decisions. For a decentralized architecture, the models are trained and executed locally or with minimal information exchange in transport networks. Hybrid architectures are in between of the centralized and decentralized architectures. Typically, ML algorithms under the hybrid architecture are trained locally by their respective training nodes, and a central server is used to orchestrate the different steps of the algorithms and coordinate all the participating nodes during the learning process. In general,

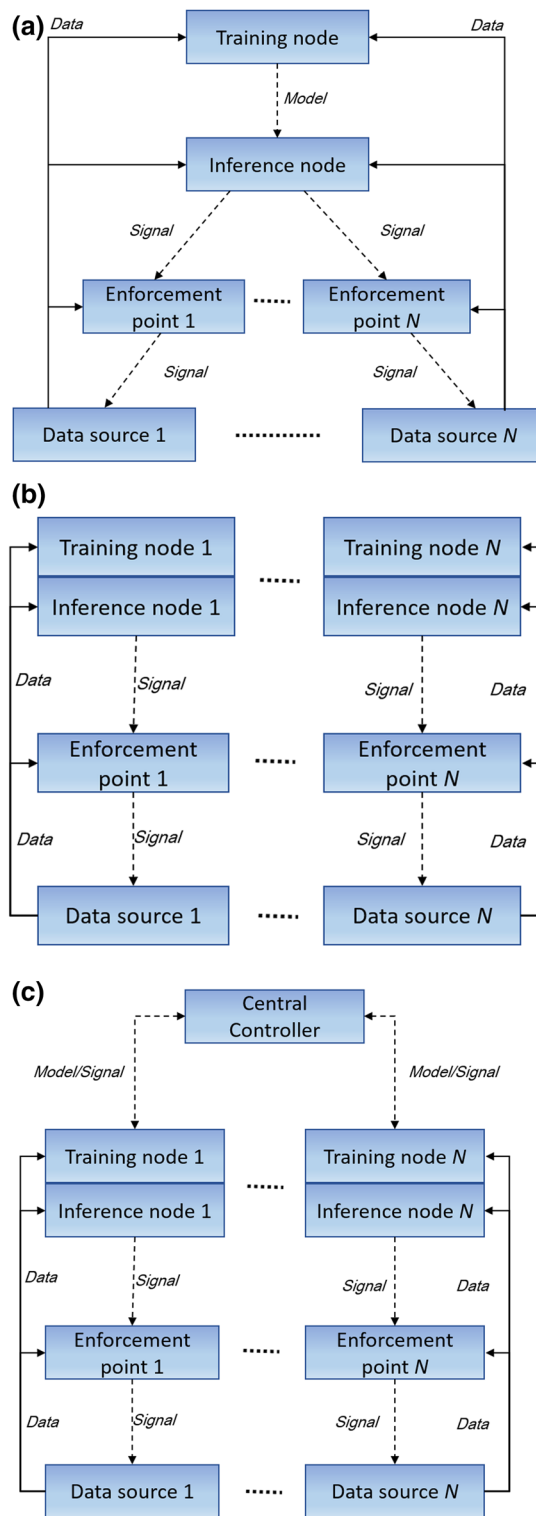


Fig. 7 Logical architecture of learning systems under three AI architectures. **a** Centralized, **b** Decentralized and **c** Hybrid

the central server will not collect data from the data sources; instead, the coordination is done by exchanging parameters (e.g. the weights and biases of a deep neural network) between the training nodes and the central server.

Based on the learning systems, we first define the following metrics for each logical node.

- At each logical node $t \in \mathcal{T}$, we denote by D_t^T the size of the training data set. In addition, we denote by Γ_t the training complexity, measured by training time or the number of training samples that the neural network needs for good test accuracy, and by C_t the algorithm convergence time (in iteration or wall clock time).
- At each inference node $i \in \mathcal{I}$, we denote by D_i^I the size of the inference data (input features) and by Π_i the inference time (wall clock time).
- At each data source $s \in \mathcal{S}$, we denote by d_s^T the amount of data generated for training and by d_s^I the amount of data generated for inference (per time unit).

Related to these metrics, we define metrics regarding the signalling information/data exchange among the different logical nodes. The data volumes are measured per time unit.

- First, we denote by $D_{s \rightarrow t}^T$ the amount of training data that need to be delivered from data source s to training node t , and we denote by $D^T = \sum_{t \in \mathcal{T}, s \in \mathcal{S}} D_{s \rightarrow t}^T$ the total amount of data needed for training.
- Second, we denote by $D_{s \rightarrow i}^I$ the amount of data delivered from data source s to inference node i and denote by $D^I = \sum_{i \in \mathcal{I}, s \in \mathcal{S}} D_{s \rightarrow i}^I$ the total data traffic for inference purpose.
- Third, we denote by $D_{i \rightarrow e}^A$ the amount of data delivered from inference node $i \in \mathcal{I}$ to enforcement point $e \in \mathcal{E}$.

4.2.3 AI/ML deployment and performance cost

The deployment cost of an AI architecture solution can be further analysed as consisting of two major cost factors: the cost for AI/ML training, C_l , and AI/ML execution costs, C_i , as follows

$$C_{\text{tot}} = C_l + C_i \quad (1)$$

To each of the cost factors in Eq. (1), there are three main components that determine the cost of the achieved performance:

- *Computing cost, $C_{*,c}$* —It refers to the cost associated with the computing resources, i.e. the amount of CPU resources, run-time memory and dedicated hardware such as GPUs or graphics acceleration. The cost is defined by the number of resource instances used and the duration they are used.
- *Networking cost, $C_{*,n}$* —It refers to the cost associated with signalling resources required for the volume of data transferred between network nodes both for the training and execution of the data and the learned models. The bandwidth and time

of the transferring may differ depending on the nodes and the capacity of their connection.

- *Storing cost, $C_{*,s}$* —It refers to the cost associated with the storage capacity in each of the network nodes. Network nodes where training will be performed benefit by managed storage facilities such as managed discs attached to compute instances.

The overall cost in Eq. (2) can be rewritten as

$$C_{\text{tot}} = C_l + C_i = (C_{l,c} + C_{l,n} + C_{l,s}) + (C_{i,c} + C_{i,n} + C_{i,s}), \quad (2)$$

where $C_{l,c}$, $C_{l,n}$, $C_{l,s}$ and $C_{i,c}$, $C_{i,n}$, $C_{i,s}$ are the compute, communicate and store costs of the learning and the inference processes, respectively.

The logical architecture of learning systems under the three AI architectures is purely data-driven, and any deployment cost values can be estimated as functions of the data sets involved. Let $c_l(D^T)$ and $c_i(D^I)$ define the learning and inference cost functions of the training and inference data sets, respectively. The deployment costs values using the data sets of the learning system nodes, as defined above, can be given by

$$C_l = c_l(D^T) = c_l \left(\sum_{t \in T, s \in S} D_{s \rightarrow t}^T \right), \quad (3)$$

$$C_i = c_i(D^I) = c_i \left(\sum_{i \in I, s \in S} D_{s \rightarrow i}^I \right). \quad (4)$$

As Eqs. (3) and (4) express, the volume and the transferring of the data sets directly determine the deployment cost for each one of AI architectures. In more specific terms, an analysis of the deployment cost tendencies can be summarized as follows:

- In a *centralized AI architecture*—the central deployment of the training and inference node implies, albeit affordable, high compute and store cost at the central server as well as a high networking cost, $C_{l,n}$, at the access points for transferring the training data set to the central server.
- In a *decentralized AI architecture*—the distributed deployment of the training and inference node implies no networking cost for the training and occasionally negligible cost for the inference process. The most dominant deployment cost in this AI architecture manifestation is related to the compute and store cost at the access points for both training $C_{l,c}$, $C_{l,s}$, and inference $C_{i,c}$, $C_{i,s}$ purposes.
- In a *hybrid AI architecture*—the deployment cost is distributed between the computing and the storing cost and the networking cost among and across the network nodes more equally. Consequently, the compute and store cost requirements at the access points are lower and the network cost higher than that of the distributed deployment cost for both training and inference procedures.

In this performance evaluation framework, the performance cost of an AI architecture solution is defined in absolute values as the ratio of the gains over the cost. In an alternative relative definition, the performance cost can be defined comparatively to a

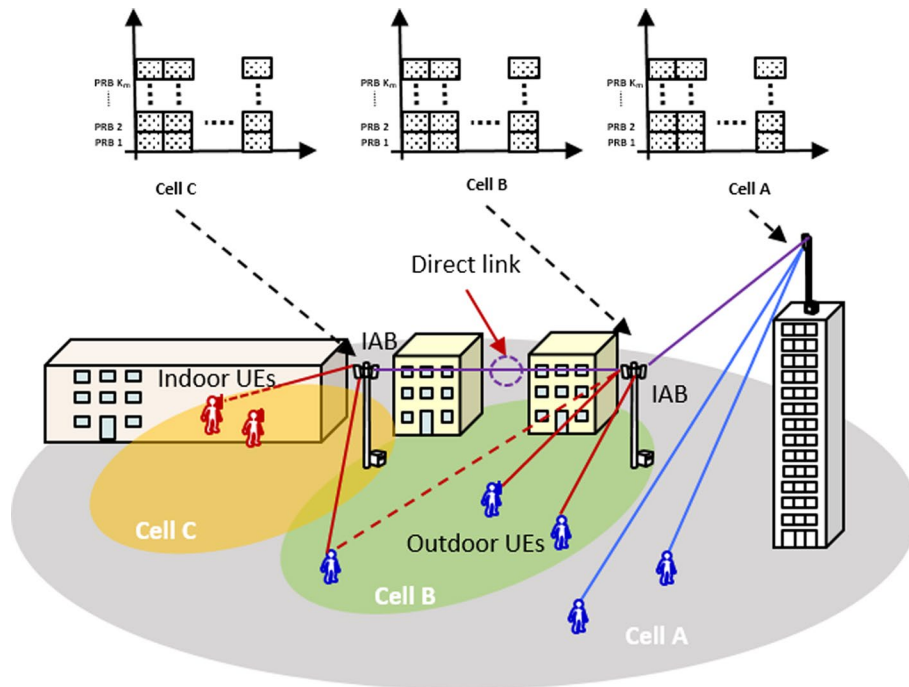


Fig. 8 A network with $|\mathcal{C}| = 1$ macro-cell and $|\mathcal{R}| = 2$ IABs

baseline solution as the difference of the achieved gained over the difference of total cost. Optionally, the performance cost can be also defined in absolute values as the ratio of the gains over the cost. The performance gain V_s of a certain set of one or multiple KPIs, k_1, k_2, \dots, k_n , normalized by the cost can be given by

$$V_s^o = \frac{o_s(k_1, k_2, \dots, k_n)}{C_s} \tag{5}$$

where $o_s(k_1, k_2, \dots, k_n)$ is an objective function of the suggested solution, and C_s is its total cost. Similarly, the cost of a suggested solution normalized by the performance gain is given by

$$V_s^c = \frac{1}{V_s^o} = \frac{C_s}{o_s(k_1, k_2, \dots, k_n)} \tag{6}$$

The normalized cost can be used when comparing a new devised solution a to a benchmark solution r . Such a relative comparison can be represented by the ratio R of the new devised solution's normalized performance V_s^o or cost V_s^c over the reference normalized performance V_r^o or cost V_r^c as follows

$$R(s, r) = \frac{V_s^o}{V_r^o} = \frac{o_s(k_1, k_2, \dots, k_n)/C_s}{o_r(k_1, k_2, \dots, k_n)/C_r} = \frac{o_s(k_1, k_2, \dots, k_n) \cdot C_r}{o_r(k_1, k_2, \dots, k_n) \cdot C_s} \tag{7}$$

where $o_r(k_1, k_2, \dots, k_n)$ and C_r are the achieved performance and total cost of the benchmark solution, respectively.

In a more general framework, multiple objective functions may be considered and the performance versus deployment cost can be calculated as a combined, potentially weighted, function of multiple use case AI optimization architecture solutions.

5 Use case analysis

In this section, we propose the ML-based algorithms with the centralized, completely decentralized and hybrid AI architectures, respectively. The three learning systems solve a multi-hop multi-link (MHML) network problem that is formulated on the notion of age-of-information (AoI). Their evaluation is performed by means of simulations based on the proposed framework in Sect. 4. The use case scenario assumes the relaying, multi-connectivity and multi-hop network capabilities [75, 81–83].

5.1 System model

We consider the uplink in a HetNet scenario as illustrated in Fig. 8. We denote by $\mathcal{C} = \{1, 2, \dots, C\}$ the set of macro-cells and by $\mathcal{R} = \{1, 2, \dots, R\}$ the set of IAB relay nodes, respectively. It has to be noted that the terms IABs nodes and relay nodes are used interchangeably in the sequel of the description. The former is used to refer to IAB node both as an access and relay node, while the latter refers to IAB node as a relay node. There is a set $\mathcal{N} = \{1, 2, \dots, N\}$ of UEs served by the network; each UE is associated with one of the access nodes in $\mathcal{M} = \mathcal{C} \cup \mathcal{R}$. We denote by $\mathcal{K}_m = \{1, 2, \dots, K_m\}$ the set of physical resource blocks (PRBs) per time slot at access node m . Depending on the equipment type and configuration, the access nodes may operate on the same or different spectrum.

We consider that time is slotted, and we denote by t_j the end of the j th time slot. We denote by t_0 the end of the initial time slot. Each access node schedules one UE per each PRB at a time slot for uplink transmission, and we denote by g_j the set of UEs that are scheduled at time slot j . Clearly, $|g_j| \leq |\mathcal{N}|$. Consider that at the j th time slot UE n is scheduled by access node $m \in \mathcal{M}$, then the transmission rate from this UE to the access node m is determined by the signal-to-interference-plus-noise ratio $\gamma(n, g_j)$

$$\gamma(n, g_j) = \frac{P_n G_{nm}}{\sum_{l \in g_j, l \neq n} P_l G_{lm} I_{ln} + \sigma_m^2}, \quad (8)$$

where P_n is the transmit power of UE n , which is assumed to be fixed, and G_{nm} is the channel gain between UE n and access node m , incorporating the effects of path loss, shadowing and fading. P_l is the transmit power of UE l and I_{ln} indicates whether or not UEs l and n transmit on the same PRB that uses the same frequency. Furthermore, σ_m^2 is the noise variance at access node m .

If data are transmitted from one IAB node to another or to a macro-cell, then we assume that the achievable rate is sufficiently large such that all buffered packets in this IAB node could be delivered in one time slot. We denote by B_{nmj} the amount of queued data from UE n in access node $m \in \mathcal{M}$ at t_{j-1} . Then at the j th time slot, the amount of data from UE n delivered by a relay node $r \in \mathcal{R}$ is B_{nrj} , and the transmission rate from this relay node to its destination equals $\sum_{n \in \mathcal{N}} B_{nrj}$ per time slot.

Table 6 Key parameters for simulation setup

Network size	500 m × 500 m	Number of UEs	21
Number of access nodes	3	Number of PRBs per node	3
Inter-cell distance	150 m to 200 m	Transmit power	20 dBm
Noise power	-100 dBm	Rate function	Shannon
Channel gain	Path loss with exponent of 4, Rayleigh fading and log-normal shadowing with standard deviation of 6 dB		

Next, we introduce the notion of information value, denoted by V_{nj} , which refers to the importance of the information from UE n at the j th time slot. The value of V_{nj} is determined by UE n based on its application. To capture the worst case scenario, we consider peak ages, which are defined as the maximal points during age evolution. We define $\mathcal{J} = \{1, 2, \dots, T\}$ as the schedule horizon. Based on the network topology, we define $\Gamma_{mr} = 1$ if relay node r is in the path of access node m to its final destination, otherwise $\Gamma_{mr} = 0$. Observe that for solving the minimum weighted AoI problem one has to solve two coupled optimization problems:

- Decide the serving node for each UE;
- Compute the scheduling strategy for the user association.

In what follows, we refer to the first optimization problem as *user association* and the second one as *uplink scheduling*. Motivated by practical system constraints, it is reasonable to assume that user association decision and task scheduling are solved on different time scales, and we thus propose to solve the two problems separately.

The *user association* problem is easily solved by assigning each UE to the serving node that yields the maximal signal-to-noise (SNR) ratio, as expressed in (9).

$$\text{SNR}(n, g_j) = \frac{P_n G_{nm}}{\sigma_m^2}, \quad (9)$$

In Table 6, we summarize the key parameters of the use case scenario and in what follows we compare different AI solutions for the uplink scheduling problem.

5.2 Solution method

Following the architecture framework, the uplink scheduling problem is solved by using deep reinforcement learning (DRL) under the centralized, decentralized and the hybrid AI architectures, as described in subsequent subsections.

5.2.1 Centralized AI architecture

In the centralized AI architecture, the training and the inference are located and performed by a central unit where the training data are transmitted from the network devices, i.e. UEs and access nodes, to the central unit. Let $\Lambda^C : \mathcal{S}^C \rightarrow \mathcal{A}^C$ denote the neural network model to be trained for the centralized AI architecture, where \mathcal{S}^C and \mathcal{A}^C are the state space and the action set, respectively. The output of the model determines which UE per access node should be scheduled for the next available PRB. Thus,

the output of the model has a dimension of M , and in every time slot, the model will be used for K_m times for inference, each for one subcarrier, such that it could select K_m UEs for access node $m \in \mathcal{M}$. The state vector, reward function and the action set are defined as follows:

- *State vector, \mathcal{S}^C* —The state vector is defined based on UE and access node-related features. The feature vector of UE n at the j th time slot is defined as

$$s_n^C(j) = [\tau_n^{j-1}, \min_{i \in \mathcal{U}_n^j} \tau_{ni}, \text{avg}_{i \in \mathcal{U}_n^j} \tau_{ni}, \Theta_{n1}, \Theta_{n2}, \dots, \Theta_{nM}], \tag{10}$$

where τ_n^{j-1} is the timestamp of the last sent packet of UE n before t_j , \mathcal{U}_n^j is the set of queued packets of UE n at t_j , $\min_{i \in \mathcal{U}_n^j} \tau_{ni}$ is the time stamp of the oldest packets of UE n , $\text{avg}_{i \in \mathcal{U}_n^j} \tau_{ni}$ is the average time stamp of the queued packets of UE n at time t_j , and Θ_{nm} is the signal strength from UE n to access node m . Next we define the vector consisting of features regarding access node m on subcarrier k at the j th time slot as

$$S_{mk}^C(j) = \left[\max_{n \in \mathcal{N}^m} V_{nj} a_{nj}, \text{avg}_{n \in \mathcal{N}^m} V_{nj} a_{nj}, \{s_n^C(j) : n \in \mathcal{N}_{k\Upsilon}^{mj}\} \right], \tag{11}$$

where $\mathcal{N}_{k\Upsilon}^{mj}$ is the set of UEs that can be served by access node m on subcarrier k and with the Υ highest weighted ages, i.e. $V_{nj} a_{nj}$, at t_j . Υ is a model parameter. The state vector at time slot j is finally defined as

$$S_k^C(j) = [\max_{n \in \mathcal{N}} V_{nj} a_{nj}, \text{avg}_{n \in \mathcal{N}} V_{nj} a_{nj}, S_{1k}^C(j), S_{2k}^C(j), \dots, S_{Mk}^C(j)]. \tag{12}$$

- *Action set, \mathcal{A}^C* —For defining the action set, recall that for each access node the model chooses one action for a specific subcarrier in an iteration. Formally, the action set is $\mathcal{A}^C = \prod_{m=1}^M \mathcal{N}_{k\Upsilon}^{mj}$. For each access node $m \in \mathcal{M}$, the algorithm iterates K_m times in each time slot and thus it chooses one UE for each subcarrier. As each access node only selects one UE to transmit in one iteration, the action space has a dimension of $\Upsilon^{|\mathcal{M}|}$, and thus, it is only the $\Upsilon^{|\mathcal{M}|}$ UEs with highest weighted ages at t_j that can be chosen to be scheduled by the model.
- *Reward function, R^C* —The immediate reward at the j th time slot is defined as

$$R^C(j) = \max_{n \in \mathcal{N}} (-V_{nj}(t_j - \tau_n^j)) - \kappa^j C, \tag{13}$$

where τ_n^j is the timestamp of the last sent packet from UE n at the j th time slot, and C is a large positive number. If any of the constraints in scheduling is violated at the j th time slot, then $\kappa^j = 1$; otherwise, $\kappa^j = 0$. That is, $\kappa^j C$ is used to train the algorithm to avoid infeasible scheduling solutions.

5.2.2 Decentralized AI architecture

According to the decentralized approach, the training and inference units are located in the access nodes $m \in \mathcal{M}$ with no information exchange between each other. The

scheduling decisions are by no means coordinated rather based on local observations only. As a result, one neural network model per access node m is trained $\Lambda_m^D : \mathcal{S}_m^D \rightarrow \mathcal{A}_m^D$ where \mathcal{S}_m^D and \mathcal{A}_m^D denote the state space and the action space of access node m , respectively.

Using the notation defined in the system model and in the centralized approach, the state vector, reward function and the action space for access node m under the decentralized approach are defined as follows:

- *State vector, \mathcal{S}^D* —The state vector at the j th time slot of the m th access node is defined as

$$\mathcal{S}_{mk}^D(j) = \left[\max_{n \in \mathcal{N}^m} V_{nj} a_{nj}, \text{avg}_{n \in \mathcal{N}^m} V_{nj} a_{nj}, \left\{ s_n^D(j), n \in \mathcal{N}_{k\Upsilon}^{mj} \right\}, I_{1m}^j, \dots, I_{Mm}^j \right]. \tag{14}$$

The above definition is based on a set of feature vectors, the elements of which correspond to UE n , feature vector, s_n^D , which is defined as

$$s_n^D(j) = \left[\tau_n^{j-1}, \min_{i \in \mathcal{U}_{nj}} \tau_{ni}, \text{avg}_{i \in \mathcal{U}_{nj}} \tau_{ni}, \Theta_{nm} \right], \tag{15}$$

where τ_n^{j-1} is the timestamp of the last sent packet of UE n before t_j , \mathcal{U}_{nj} denotes the set of queued packets of UE n at t_j and Θ_{nm} is the signal strength from UE n to access node m . I_{lm}^j denotes the measured interference from the UE(s) scheduled by access node l at access node m and is computed as the average interference in the past t_s time slots.

- *Action set, \mathcal{A}^D* —The action set is $\mathcal{A}_m^D = \mathcal{N}_{k\Upsilon}^{mj}$, as each access node m only schedules one UE in each iteration.
- *Reward function, R^D* —The immediate reward at the end of the j th time slot is the maximum weighted age after the scheduling decision has been implemented, i.e.

$$R_m^D(j) = \max_{n \in \mathcal{N}^m} (-V_{nj}(t_j - \tau_n^j)) - \kappa^j C, \tag{16}$$

where τ_n^j is the timestamp of the last sent packet from UE n at the j th time slot, and C is a large positive number. If any of the constraints in scheduling is violated at the j th time slot, then $\kappa^j = 1$; otherwise, $\kappa^j = 0$.

5.2.3 Hybrid AI architecture

In the hybrid AI architecture, the access nodes can exchange information so as to improve the scheduling solutions compared to the decentralized architecture. To reduce the size of the signalling, we exchange a latent space $\Omega_{m,m'} \in \mathcal{R}^{d_m}$ of the state S_m , one for each neighbouring node m' that is created by the encoder $F_m^{AE,E}$ of a trained autoencoder.

The encoder of the trained autoencoder is replicated to obtain a pre-trained encoder $F_{m,m'}^E = F_m^{AE,E}$ for each neighbour of access node m , and the $\Omega_{m,m'} = F_{m,m'}^E(S_m)$ comprises the latent space to be sent to the neighbouring node m' as an input for scheduling. The dimension d_m of the latent space is a meta-parameter that is fixed. The

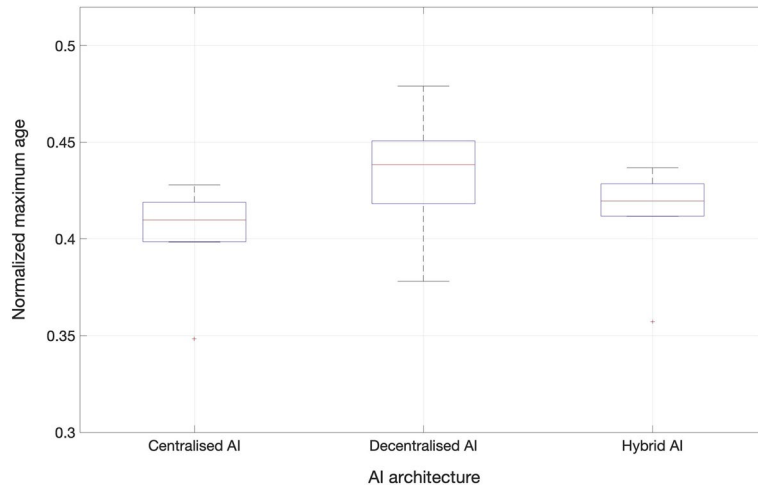


Fig. 9 Comparison of the three learning approaches based on maximum age reduction over PF systems

Table 7 Performance comparison for the three learning systems

Learning system	Centralized AI	Decentralized AI	Hybrid AI
Average age reduction over PF	60%	58%	59%
Maximum age reduction over PF	65%	62%	64%
Inter access node traffic for training	120 KBytes/s per UE	n/a	30 KBytes/s per UE
Inter access node signal for synch	1 Kbyte/s per UE	1 Kbyte/s per UE	1 KByte/s per UE
Average/Maximum age reduction gain (%-e) per unit cost (1 KBps)	0.50/0.54	58/62	1.90/2.06
Average/Maximum age reduction cost (KBps) per percentage gain (%-e)	2.02/1.86	0.02/0.02	0.53/0.48

neural network model for access node m with the hybrid AI architecture is denoted by $\Lambda_m^H : \mathcal{S}_m^H \rightarrow \mathcal{A}_m^H$, where \mathcal{S}_m^H and \mathcal{A}_m^H correspond to the state vector and the action set of access node m , respectively.

- *State vector, \mathcal{S}^H* —The state vector of the hybrid AI architecture at the j th time slot is defined as

$$\mathcal{S}_m^H(j) = \left[\mathcal{S}_m^D(j), \Omega_{m_1,m}, \Omega_{m_2,m}, \dots, \Omega_{m',m} \right], \tag{17}$$

where $\mathcal{S}_m^D(j)$ is the state vector under the decentralized approach, as defined in (14), and $\{m_1, m_2, \dots, m'\} \subset \mathcal{M}$ is the set of neighbouring nodes of m .

- *Action set, \mathcal{A}^H* —In a similar manner as with decentralized, scheduling decisions are made per subcarrier iteration, implying an action set defined as $\mathcal{A}_m^H = \mathcal{N}_{k^*}^{mj}$.
- *Reward function, R^H* —Equivalently to the decentralized approach, the immediate reward at the end of the j th time slot is similar to the maximum weighted age of information definition in (16), i.e.

$$R_m^H(j) = \max_{n \in \mathcal{N}^m} (-V_{nj}(t_j - \tau_n^j)) - \kappa^j C. \tag{18}$$

Table 8 Comparison ratio based on the average age reduction and the associated cost of the three learning systems

Ratio $R(s_X, r_Y)$	Centralized AI V_{r_C}	Decentralized AI V_{r_D}	Hybrid AI V_{r_H}
Centralized AI V_{s_C}	1.00	116.97	3.84
Decentralized AI V_{s_D}	0.01	1.00	0.03
Hybrid AI V_{s_H}	0.26	30.47	1.00

5.3 Evaluation results

Our evaluation studies show that the different AI architecture alternatives achieve different gains and pose different deployment costs when considering the deployment and selection of the AI/ML solution. The three AI architecture solutions have been evaluated by means of average and maximum age reduction over a proportional fair (PF) baseline scenario, where users are scheduled based on a PF-scheduling discipline. Figure 9 shows the maximum age reduction of the three AI architectures as simulated by ten different network instances.

Table 7 provides a comparative summary of the performance and cost of the three learning systems. We observe that DRL with all three architectures could significantly improve the freshness of information compared to the proportional fair algorithm. Among the three AI architectures, as expected, the centralized one achieves the best performance, at the price of notable inter-node traffic and a long training time, as well as potential privacy and security issues during data exchange. On the opposite, under the decentralized architecture the models could be trained with insignificant data exchange among access nodes, but the performance improvement is lower than that for the centralized architecture. The hybrid approach trades off performance for training data exchange, and it achieves a decent performance in age reduction with only around a quarter of inter-node traffic data for training compared to the centralized approach.

For the centralized approach, in practice the training will be implemented on a central server, which could be more powerful than our hardware, yet initial training of the model would be computationally intensive. The same concern applies to the decentralized and the hybrid architectures, and the training of the scheduler takes a considerable amount of computing resources. Naturally, the performance, the training time and the inference time also depend on the training data, the neural network structure and hyper-parameters, but the training time is expected to be significant in lack of specialized hardware (e.g. tensor processing unit).

Although computation intensity could be defined as the cost of each learning approach, it is subject to the hardware used and the training times showed no significant differences. For a more hardware-agnostic scenario, we have defined the cost by means of signalling between the nodes. The signalling consists of two parts: the data exchanged required for the learning process and the inference imposed synchronization signalling between the nodes. While the synchronization signalling has been estimated not to exceed 1 KByte per second, the cost of the learning between the three different AI learning architectures differs significantly. The centralized AI solution requires up to 120 KBytes per second between any two nodes, while the hybrid AI approach requires only 30 KBytes per second. Table 7 reports on cost normalized maximum age reduction where the percentage gains for a signalling cost

of 1KBps for the centralized, hybrid and decentralized AI are in the order of 0.54, 2.06 and 62%, while the cost per percentage reduction gain is 1.86, 0.48 and 0.02 KBps, respectively.

As a result, the centralized AI architecture approach demonstrates the highest cost per gained percent, which according to Table 8 corresponds to an 116 times higher value over the decentralized approach. By inspecting Table 8, it is evident that the efficiency of hybrid AI approach in terms of cost normalized performance tends to be higher than the centralized and significantly closer to that of the decentralized.

6 Conclusions

AI/ML techniques have been seen to be promising in many applications in communication networks, and it is no doubt that AI/ML would be an inherent element in 5G systems and beyond. This path gives birth to the challenging topic on how to integrate AI into the 5G systems such that we could leverage the advantages of ML techniques to optimize and improve the B5G networks. In this article, we present the background and the challenges that the current communication systems face and reviewed the related work of ML and applications in communication networks. Then, we have proposed an AI architecture and performance evaluation framework for the deployment of the AI/ML solution in B5G networks. The suggested framework defines three AI architectures alternatives, a centralized, a completely decentralized and a hybrid AI architecture. We have further identified the logical AI functions and their mapping to the B5G RAN architecture and analysed the associated deployment cost factors in terms of compute, communicate and store costs. It is shown, by means of simulation-based evaluations, that the different AI architecture alternatives pose different signalling costs when considering the deployment and selection of the AI/ML solution.

In our future work, we will develop ML algorithms under the hybrid AI architecture and evaluate the performance in demanding heterogeneous network scenarios.

Abbreviations

3GPP	Third Generation Partnership Project
5G	5th generation
5GC	5G core
AF	Application function
AI	Artificial intelligence
AMF	Access and mobility management function
AUSF	Authentication server function
AP	Access point
B5G	Beyond 5th generation
BBU	Baseband unit
CN	Core network
CP	Control plane/coordination point
CPU	Central process unit
CU	Central unit
DN	Data network
DNN	Deep neural network
DQL	Deep Q-learning
DRL	Deep reinforcement learning
E-UTRA	Evolved universal terrestrial radio access
E2E	End-to-end
EM	Expectation maximization
eMBB	enhanced mobile broadband
eNB	eNodeB
gNB	gNodeB
GPU	Graphical process unit
HMM	Hidden Markov model
IAB	Integrated access backhaul

ICA	Independent component analysis
ITU	International Telecommunication Union
KNN	kth nearest neighbour
KPI	Key performance indicator
LTE	Long-term evolution
MAC	Medium access control
MDP	Markov decision process
MEC	Mobile edge cloud
MIMO	Multiple input multiple output
ML	Machine learning
MMTC	Massive machine-type communications
NEF	Network exposure function
NF	Network function
NFV	Network function virtualization
NG-RAN	Next-generation radio access network
NR	New radio
NRF	Network repository function
NSSF	Network slice selection function
NWDAF	Network data analytics function
O-RAN	Open radio access network
PCA	Principal component analysis
PCF	Policy control function
PDCP	Packet data convergence protocol
PF	Proportional fair
PRB	Physical resource block
RAN	Radio access network
RL	Reinforcement learning
RLC	Radio link control
RRC	Radio resource control
RRM	Radio resource management
SDAP	Service data adaptation protocol
SDN	Software-defined networking
SEPP	Security edge protection proxy
SMF	Session management function
SNR	Signal-to-noise ratio
SVM	Support vector machines
TP	Terminal point
UDM	Unified data management
UDR	Unified data repository
UDSF	Unstructured data storage function
UE	User equipment
UP	User plane
UPF	User plane function
URLLC	Ultra-reliable low-latency communications

Author contributions

All authors have contributed to the conception, design of the work; the acquisition, analysis and interpretation of data; and have edited and approved the submitted version. The second author has in addition created the software used for the research study experiments of the use case scenario. All authors read and approved the final manuscript.

Funding

Open access funding provided by Royal Institute of Technology. This work has been funded by Huawei Technologies Sweden AB under the Huawei Innovation Research Program (HIRP) agreement HO2029042720036P181.

Availability of data and materials

The datasets generated and/or analysed during the current study are not publicly available due to license agreement restrictions, but data related to the implementation of use case scenario can be made available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 17 May 2022 Accepted: 25 August 2022

Published online: 24 September 2022

References

1. T.S. Rappaport, Y. Xing, G.R. MacCartney Jr., A.F. Molisch, E. Mellios, J. Zhang, Overview of millimeter wave communications for fifth-generation (5G) wireless networks—with a focus on propagation models. *IEEE Trans. Antennas Propag.* **65**(12), 6213–6230 (2017)
2. A. Mchangama, J. Ayadi, V.P.G. Jiménez, A. Consoli, MmWave massive MIMO small cells for 5G and beyond mobile networks: an overview, *12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 1–6 (2020)
3. C. Saha, H.S. Dhillon, On load balancing in millimeter wave HetNets with integrated access and backhaul, in *IEEE Global Communications Conference (GLOBECOM)* (2019), pp. 1–6
4. U. Gustavsson, P. Frenger, C. Fager, T. Eriksson, H. Zirath, F. Dielacher, C. Studer, A. Pärssinen, R. Correia, J.N. Matos, D. Belo, N.B. Carvalho, Implementation challenges and opportunities in beyond-5G and 6G communication. *IEEE J. Microw.* **1**(1), 86–100 (2021)
5. ITU-T FG-ML5G-ARC5G, Unified architecture for machine learning in 5G and future networks, in *Telecommunication Standardization Section of ITU* (2019)
6. O-RAN Working Group 2. AI/ML workflow description and requirements
7. FuTURE Forum. Wireless Big Data for Smart 5G
8. 5G PPP Architecture Working Group. View on 5G Architecture, Version 4.0, October (2021)
9. 3GPP Technical Specification Group Core Network and Terminals. TS 29.520 5G System; Network Data Analytics Services; Stage 3 (Release 17). ver.V17.7.0 June (2022)
10. 3GPP Technical Specification Group Services and System Aspects. TS 23.501 System architecture for the 5G System (5GS) (Release 17), ver.17.5.0, June (2022)
11. 3GPP Technical Specification Group Radio Access Network. TS 38.401 NG-RAN Architecture description (Release 17), ver.17.1.1, July (2022)
12. 3GPP Technical Specification Group Radio Access Network. TS 38.300 NR; NR and NG-RAN Overall Description (Release 17), ver.17.1.0, July (2022)
13. 3GPP Technical Specification Group Radio Access Network. TS 36.331 NR; Radio Resource Control (RRC); Protocol specification (Release 17). ver.17.1.0, July (2022)
14. 3GPP Technical Specification Group Services and System Aspects. TS 38.331 Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (Release 17). ver.17.1.0, June (2022)
15. 3GPP Technical Specification Group Radio Access Network. TS 38.351 E-UTRA and NR; Service Data Adaptation Protocol (SDAP) specification (Release 17), ver.17.0.0, March (2022)
16. 3GPP Technical Specification Group Radio Access Network. TS 38.323 NR; Packet Data Convergence Link Protocol (PDCP) specification (Release 17), ver.17.1.0, June (2022)
17. 3GPP Technical Specification Group Radio Access Network. TS 38.322 NR; Radio Link Control (RLC) protocol specification (Release 17), ver.17.1.0, June (2022)
18. 3GPP Technical Specification Group Radio Access Network. TS 38.321 NR; Medium Access Control (MAC) protocol specification (Release 17), ver.17.1.0, June (2022)
19. 3GPP Technical Specification Group Radio Access Network. TS 38.322 NR; Physical layer; General description (Release 17), ver.17.0.0, December (2021)
20. 3GPP Technical Specification Group Radio Access Network. TS 38.470 NG-RAN; F1 general aspects and principles (Release 17), ver.17.1.0, June (2022)
21. 3GPP Technical Specification Group Radio Access Network. TS 38.460 NG-RAN; E1 general aspects and principles (Release 17), ver.17.0.0, April (2022)
22. 3GPP Technical Specification Group Radio Access Network. TS 38.420 NG-RAN; Xn general aspects and principles (Release 17), ver.17.1.0, June (2022)
23. 3GPP Technical Specification Group Radio Access Network. TS 38.410 NG-RAN; NG general aspects and principles (Release 17), ver.17.1.0, June (2022)
24. I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, H. Flinck, Network slicing and softwarization: a survey on principles enabling technologies and solutions. *IEEE Commun. Surv. Tutor.* **20**(3), 2429–2453 (2018)
25. R. Shubbar, M. Alhisanawi, A. Abdhassan, M. Ahamdi, A comprehensive survey on software-defined network controllers, in *Next Generation of Internet of Things Lecture Notes in Networks and Systems*, vol. 201, ed. by R. Kumar, B.K. Mishra, P.K. Pattnaik (Springer, Singapore, 2021), pp.1–33
26. R. Jain, S. Paul, Network virtualization and software defined networking for cloud computing—a survey. *IEEE Commun. Mag.* **51**(11), 24–31 (2013)
27. J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J.J. Ramos-Munoz, J.M. Lopez-Soler, A survey on 5G usage scenarios and traffic models. *IEEE Commun. Surv. Tutor.* **22**(2), 905–929 (2020)
28. R. Boutaba, M.A. Salahuddin, N. Limam, S. Ayoubi, M. Shahriar, F.E. Solano, O.M. Caicedo, A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *J Internet Serv. Appl.* **9**, 16 (2018). <https://doi.org/10.1186/s13174-018-0087-2>
29. K. Zia, N. Javed, M.N. Sial, S. Ahmed, A.A. Pirzada, F. Pervez, A distributed multi-agent RL-based autonomous spectrum allocation scheme in D2D enabled multi-tier HetNets. *IEEE Access* **7**, 6733–6745 (2019)
30. C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, L. Hanzo, Machine learning paradigms for next-generation wireless networks. *IEEE Wirel. Commun.* **24**(2), 98–105 (2017)
31. S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd edn. (Prentice Hall Press, Upper Saddle River, 2009), pp.649–789
32. B.K. Donohoo, C. Ohlsen, S. Pasricha, Y. Xiang, C. Andersson, Context-aware energy enhancements for smart mobile devices. *IEEE Trans. Mob. Comput.* **13**(8), 1720–1732 (2014)
33. C. Wen, S. Jin, K. Wong, J. Chen, P. Ting, Channel estimation for massive MIMO using gaussian-mixture Bayesian learning. *IEEE Trans. Wireless Commun.* **14**(3), 1356–1368 (2015)
34. A. Assra, J. Yang, B. Champagne, An EM approach for cooperative spectrum sensing in multiantenna CR networks. *IEEE Trans. Veh. Technol.* **65**(3), 1229–1243 (2016)

35. K.W. Choi, E. Hossain, Estimation of primary user parameters in cognitive radio systems via hidden Markov model. *IEEE Trans. Signal Process.* **61**(3), 782–795 (2013)
36. C. Yu, K. Chen, S. Cheng, Cognitive radio network tomography. *IEEE Trans. Veh. Technol.* **59**(4), 1980–1997 (2010)
37. C. Zhang, P. Patras, H. Haddadi, Deep learning in mobile and wireless networking: a survey. *IEEE Commun. Surv. Tutor.* **21**(3), 2224–2287 (2019)
38. M. Xia, Y. Owada, M. Inoue, H. Harai, Optical and wireless hybrid access networks: design and optimization. *IEEE/OSA J. Opt. Commun. Netw.* **4**(10), 749–759 (2012)
39. H. Nguyen, G. Zheng, R. Zheng, Z. Han, Binary inference for primary user separation in cognitive radio networks. *IEEE Trans. Wireless Commun.* **12**(4), 1532–1542 (2013)
40. C. Qiu, Z. Hu, Z. Chen, N. Guo, R. Ranganathan, S. Hou, G. Zheng, Cognitive radio network for the smart grid: experimental system architecture, control algorithms, security, and microgrid testbed. *IEEE Transactions on Smart Grid* **2**(4), 724–740 (2011)
41. N.C. Luong, D.T. Hoang, S. Gong, D. Niyato, P. Wang, Y. Liang, D.I. Kim, Applications of deep reinforcement learning in communications and networking: a survey. *IEEE Commun. Surv. Tutor.* **21**(4), 3133–3174 (2019)
42. Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: concept and applications. *ACM Trans. Intell. Syst. Technol.* **67**(10), 1–19 (2019)
43. S.P. Sotirioudis, A.D. Boursianis, S.K. Goudos, K. Siakavara, an ensemble learning approach. *IEEE Trans. Antennas Propag. From Spatial Urban Site Data to Path Loss Prediction* 1–1 (2021)
44. N. Moraitis, L. Tsipi, D. Vouyioukas, A. Gkioni, S. Louvros, On the assessment of ensemble models for propagation loss forecasts in rural environments. *IEEE Wirel. Commun. Lett.* **11**(5), 1097–1101 (2022)
45. S.K. Goudos, G. Athanasiadou, Application of an ensemble method to UAV power modeling for cellular communications. *IEEE Antennas Wirel. Propag. Lett.* **18**(11), 2340–2344 (2019)
46. J. McCoy, A. Rawal, D.B. Rawat, B.M. Sadler, Ensemble Deep learning for sustainable multimodal UAV classification. *IEEE Transactions on Intelligent Transportation Systems.* 1–10 (2022). <https://doi.org/10.1109/TITS.2022.3170643>
47. R. Sahay, C.G. Brinton, D.J. Love, A deep ensemble-based wireless receiver architecture for mitigating adversarial attacks in automatic modulation classification. *IEEE Trans. Cogn. Commun. Netw.* **8**(1), 71–85 (2022)
48. S. Imtiaz, G.P. Koudouridis, J. Gross, On the feasibility of coordinates-based resource allocation through machine learning, in *IEEE Global Communications Conference (GLOBECOM)*, pp 1–7 (2019)
49. S. Imtiaz, S. Schiessl, G.P. Koudouridis, J. Gross, Coordinates-based resource allocation through supervised machine learning. *IEEE Trans. Cogn. Commun. Netw.* **7**(4), 1347–1362 (2021)
50. Y. Sun, S. Zhou, J. Xu, EMM: energy-aware mobility management for mobile edge computing in ultra dense networks. *IEEE J. Sel. Areas Commun.* **35**(11), 2637–2646 (2017)
51. Z. Li, C. Wang, C. Jiang, User association for load balancing in vehicular networks: an online reinforcement learning approach. *IEEE Trans. Intell. Transp. Syst.* **18**(8), 2217–2228 (2017)
52. C. Dhahri, T. Ohtsuki, Q-learning cell selection for femtocell networks: single- and multi-user case, in *Proceedings of the 2012 IEEE Global Communications Conference (GLOBECOM)* (2012), pp. 4975–4980
53. T. Kudo, T. Ohtsuki, in *Proceedings of the IEEE 78th Vehicular Technology Conference (VTC Fall)* (2013), pp. 1–5
54. V. Sciancalepore, X. Costa-Perez, A. Banchs, RL-NSB: reinforcement learning-based 5G network slice broker. *IEEE/ACM Trans. Netw.* **27**(4), 1543–1557 (2019)
55. X. Chen, Z. Zhao, C. Wu, M. Bennis, H. Liu, Y. Ji, H. Zhang, Multi-tenant cross-slice resource orchestration: a deep reinforcement learning approach. *IEEE J. Sel. Areas Commun.* **37**(10), 2377–2392 (2019)
56. X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, M. Bennis, Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. *IEEE Internet Things J.* **6**(3), 4005–4018 (2019)
57. Q. He, A. Moayyedi, G. Dán, G.P. Koudouridis, P. Tengkvist, A meta-learning scheme for adaptive short-term network traffic prediction. *IEEE J. Sel. Areas Commun.* **38**(10), 2271–2283 (2020)
58. D. Bega, M. Gramaglia, M. Fiore, A. Banchs, Costa-Perez, X, in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)* (2019), pp. 280–288
59. L. Ale, N. Zhang, H. Wu, D. Chen, T. Han, Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network. *IEEE Internet Things J.* **6**(3), 5520–5530 (2019)
60. Y. He, N. Zhao, H. Yin, Integrated networking, caching, and computing for connected vehicles: a deep reinforcement learning approach. *IEEE Trans. Veh. Technol.* **67**(1), 44–55 (2018)
61. S. Boll, (Ed.), *MM '18: Proceedings of the 26th ACM International Conference on Multimedia*, 3rd ed.; Publisher: Association for Computing Machinery, New York, NY, USA, (2018) 154–196
62. M. Elsayed, M. Erol-Kantarci, H. Yanikomeroğlu, Transfer reinforcement learning for 5G-NR mm-wave networks. [arXiv:2012.04840](https://arxiv.org/abs/2012.04840) (2020)
63. Y. Yang, Y. Li, K. Li, S. Zhao, R. Chen, J. Wang, S. Ci, DECCO: deep-learning enabled coverage and capacity optimization for massive MIMO systems. *IEEE Access* **6**(1), 23361–23371 (2018)
64. Z. Xu, Y. Wang, J. Tang, J. Wang, M.C. Gursos, in *Proceedings of the IEEE International Conference on Communications (ICC)* (2017), pp. 1–6
65. B. Yin, S. Zhang, Y. Cheng, Application-oriented scheduling for optimizing the age of correlated information: a deep-reinforcement-learning-based approach. *IEEE Internet Things J.* **7**(9), 8748–8759 (2020)
66. ITU-T FG-ML5G-ARC5G, Y.3172: architectural framework for machine learning in future networks including IMT-2020, in *Telecommunication Standardization Section of ITU* (2019)
67. F.D. Calabrese, L. Wang, E. Ghadimi, G. Peters, L. Hanzo, P. Soldati, Learning radio resource management in RANs: framework, opportunities, and challenges. *IEEE Commun. Mag.* **56**(9), 138–145 (2018)
68. S. Ayoubi, N. Limam, M.A. Salahuddin, N. Shahriar, R. Boutaba, F. Estrada-Solano, O.M. Caicedo, Machine learning for cognitive network management. *IEEE Commun. Mag.* **56**(1), 158–165 (2018)
69. Telecom Infra Project AI/ML Project Group, AI and applied machine learning (2017)
70. TIP Open RAN MoU signatories, *Open RAN Technical Priority Document* (Deutsche Telekom, Orange, Telefónica, TIM, Vodafone, 2021)

71. 5G PPP Technology Board AI and ML–Enablers for Beyond 5G Networks. 5GPPP White Paper (2021)
72. A. Zappone, M. Di Renzo, M. Debbah, Wireless networks design in the era of deep learning: model-based, AI-based, or both? *IEEE Trans. Commun.* **67**(10), 7331–7376 (2019)
73. C. Wang, M. Di Renzo, S. Stanczak, S. Wang, E.G. Larsson, Artificial intelligence enabled wireless networking for 5G and beyond: recent advances and future challenges. *IEEE Wirel. Commun.* **27**(1), 16–23 (2020)
74. M. Polese, R. Jana, V. Kounev, K. Zhang, S. Deb, Zorzi, M. Machine Learning at the Edge: A Data-Driven Architecture With Applications to 5G Cellular Networks. in *IEEE Transactions on Mobile Computing*, vol. 20, pp. 3367–3382 (2021)
75. 3GPP Technical Specification Group Radio Access Network. TS 38.174 NR; Integrated access and backhaul radio transmission and reception (Release 17), ver.17.1.0, June (2022)
76. 3GPP Technical Specification Group Services and System Aspects. TS 38.331 Management and orchestration; 5G end to end Key Performance Indicators (KPI) (Release 17). ver.17.7.0, June (2022)
77. 3GPP Technical Specification Group Services and System Aspects. TS 36.314 Evolved Universal Terrestrial Radio Access (E-UTRA); Layer 2—Measurements (Release 16). ver.16.0.0, July (2020)
78. 3GPP Technical Specification Group Services and System Aspects. TS 36.314 New Radio (NR); Layer 2—Measurements (Release 16). ver.16.4.0, September (2021)
79. 3GPP Technical Specification Group Services and System Aspects. TS 28.552 Management and orchestration; 5G performance measurements (Release 17). ver.17.5.0, December (2021)
80. 3GPP Technical Specification Group Services and System Aspects. TS 32.425 Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Telecommunication management; Performance Management (PM); Performance measurements (Release 17). ver.17.1.0, June (2021)
81. 3GPP Technical Specification Group Radio Access Network. TS 37.340 Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Multi-connectivity (Release 17), ver.17.1.0, June (2022)
82. 3GPP Technical Specification Group Radio Access Network. TS 38.340 NR; Backhaul Adaptation Protocol (BAP) specification (Release 17), ver.17.1.0, July (2022)
83. 3GPP Technical Specification Group Radio Access Network. TS 38.351 NR; Sidelink Relay Adaptation Protocol (SRAP) specification (Release 17), ver.17.1.0, June (2022)
84. F.D. Calabrese, P. Frank, E. Ghadimi, U. Challita, P. Soldati, enhancing RAN performance with AI. *Ericsson Technol. Rev.* **101**, 36–46 (2020)

Publisher's note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
