

Applicability Domains of Machine Learning in Next Generation Radio Access Networks

R. Ferrús, O. Sallent, J. Pérez-Romero, R. Agustí

Dept. of Signal Theory and Communications
Universitat Politècnica de Catalunya (UPC),
Barcelona, Spain

[ferrus,sallent,jorperez,ramon]@tsc.upc.edu

Abstract— The Radio Access Network (RAN) is envisaged to undergo a significant transformation in the context of 5G and beyond mobile communications systems. One of the driving forces behind this transformation is the applicability of Machine Learning (ML) techniques. Taking as a reference the high-level architecture for a next generation RAN proposed by the Open RAN Alliance, this paper identifies the applicability domains where ML techniques can play a relevant role. For each domain, namely radio physical layer processing, Medium Access Control (MAC) scheduling, near-real time Radio Resource Management (RRM), RAN data analytics and RAN operational automation, the paper discusses the specific functionalities that can benefit from the application of ML and analyses the key considerations and challenges that need to be addressed when developing ML-based solutions, given the particular characteristics of the mobile radio environment.

Keywords—Next Generation Radio Access Network, Machine Learning, Radio Resource Management, RAN Management

I. INTRODUCTION

As mobile traffic increases and new usage scenarios are addressed (e.g. industrial networks, vehicular communications) mobile networks and the equipment that runs them must become more software-driven, agile, flexible, intelligent and resource efficient to simplify deployments and, ultimately, to drive down CAPEX and OPEX. The advent of Software Defined Networking (SDN) and Network Function Virtualization (NFV) technologies has already brought significant changes in this direction in the Core Network (CN), with less impact so far in the Radio Access Network (RAN). In this context, initiatives such as the Open RAN (O-RAN) Alliance, led by major wireless operators around the globe, are pushing towards evolving the RAN and making it more open and smarter than previous generations [1]. This evolution is sustained in two main principles: *openness*, which entails the adoption of open interfaces, open source software and hardware reference designs for achieving a more modular, disaggregated, multi-vendor RAN implementation; and *intelligence*, necessary to cope with the increasing complexity by means of new learning based technologies to automate operational network functions and reduce traditional human intensive means.

Indeed, the exploitation of Artificial Intelligence (AI) and, more specifically, Machine Learning (ML) techniques, is anticipated to be a game-changer for network operators at all levels, ranging from top business, service and network management levels (e.g. customer care management, service

fault management, network performance management) down to the level of driving the operation of specific functions embedded within the RAN nodes.

In this context, this paper attempts to provide a comprehensive overview of the main applicability domains of ML in the next generation RAN. The identification of these domains allows us delineating the different approaches and challenges that exist across the whole chain of RAN components and associated operations support systems.

The rest of the paper is organized as follows. Section II describes the architecture of a next generation RAN and identifies the applicability domains of ML in this architecture. Then, subsequent Sections III to VI address in detail each of these domains, discussing the applicability areas of ML and including the key challenges and considerations derived from the specific characteristics of wireless networks. Finally, Section VII summarizes the conclusions reached.

II. NEXT GENERATION RAN AND ML APPLICABILITY DOMAINS

Fig. 1 illustrates the high-level architecture of a next generation RAN along with the main components of the Operations Support Systems (OSS) needed for RAN orchestration and management and RAN engineering. The RAN architecture depicted is based on the O-RAN reference architecture [2], which establishes a foundation for building a virtualized RAN on open hardware, based on well defined, standardized interfaces and with embedded AI-powered radio control. The O-RAN architecture is, in turn, compliant and complementary to the standardized architecture promoted by the Third Generation Partnership Project (3GPP) for both Long Term Evolution (LTE) and 5G New Radio (5G NR) interfaces [3][4].

According to this architecture, the processing of the radio protocol stack from the Radio Resource Control (RRC) to the physical (PHY) layer is distributed between a Radio Unit (RU), a Distributed Unit (DU) and a Central Unit (CU). Starting from the bottom of Fig. 1, the RU, which is the node attached to or integrating the antennas, hosts the RF processing (e.g. transceivers, analog beamforming, power amplifier) as well as the lower part of the PHY layer functionality, referred to as Low-PHY (e.g. digital beamforming, Fast Fourier Transform (FFT)/inverse FFT processing). Connected to the RU via the so called “Open Fronthaul” interface, the DU is in charge of the High-PHY layer (e.g. channel coding, scrambling,

modulation) processing, Medium Access Control (MAC) and Radio Link Control (RLC). On top of the DU, the CU hosts the upper layers of the radio protocol stack, i.e. RRC, Packet Data Convergence Protocol (PDCP) and Service Data Adaptation Protocol (SDAP) layers. Of note is that according to the O-RAN architecture, the CU could be a multi-Radio Access Technology (multi-RAT) unit, i.e., handle simultaneously both LTE DUs and 5G NR DUs. In general terms, the RU and DU nodes require hardware specific implementations, while the multi-RAT CU node can be executed in a NFV infrastructure (NFVI).

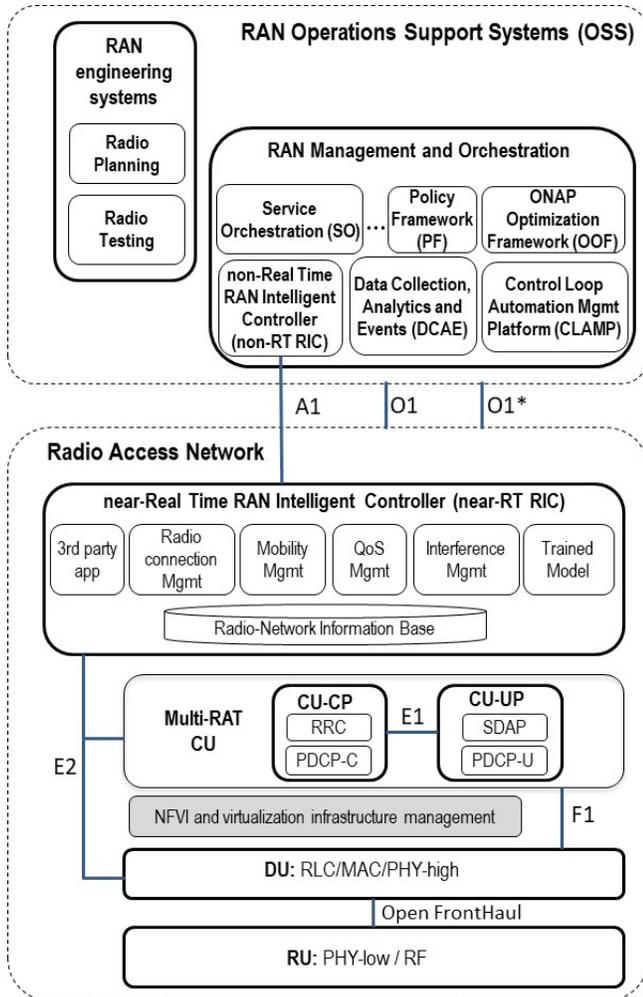


Figure 1. High-level architecture of a next generation RAN along with the accompanying OSS main components

The control of the radio protocol stack processing within the CU and DU nodes is performed by the so-called near-real-time RAN Intelligent Controller (near-RT RIC) via the E2 interface. Such interface enables feeding data, including various RAN measurements, to the near-RT RIC as well as it allows the near-RT RIC to initiate configuration commands directly to the CU/DU. Internally, the near-RT RIC is in charge of Radio Resource Management (RRM) functions such as radio connection, mobility, Quality of Service (QoS) and interference management. It also allows for flexible on-boarding of third-party control applications

and embeds a database called the Radio-Network Information Base (R-NIB) which captures the near real-time state of the underlying network.

Focusing now on the OSS functions necessary for the operation and management of the RAN nodes (i.e. RU/DU/CU/near-RT RIC), there is an important convergence within the mobile industry towards the adoption of mainstream network orchestration and management platforms such as, for example, Open Network Automation Platform (ONAP) [5][6]. ONAP provides capabilities for the design, creation, orchestration, monitoring, and life cycle management of softwarized networks and services, including support for network slicing management. In this respect, ONAP integrates diverse subsystems such as: Service Orchestration (SO), to automatically instantiate services and network components; Data Collection, Analytics and Events (DCAE), to collect, transform and store data as necessary for analysis; Closed Loop Automation Management Platform (CLAMP), to provide the mechanisms for designing and managing control loops; Policy Framework (PF), to create and manage policies that underlie the operation of other ONAP subsystems; and ONAP Optimization Framework (OOF), to provide a policy-driven approach for creating and running optimization applications for a broad range of use cases, including Self-Organizing Network (SON) functions.

In addition to these core subsystems, the ONAP platform is designed to easily integrate extensions in the form of specific controllers needed to establish the coupling between the platform and the underlying managed network. These controllers are actually the components that execute the configuration, real-time policies, and control the state of the managed functionality. In this regard, the type of controller to be integrated in the RAN management and orchestration platform is the so called non-real-time RIC (non-RT RIC), whose scope is defined as part of the O-RAN architecture. The non-RT RIC interacts with the near-RT RIC via the A1 interface, whose standardization is currently pursued within the O-RAN Alliance [7]. Through A1 interface, the non-RT RIC provides services for policy management, ML model management (e.g. sending of trained ML models to the near-RT RIC) and delivery of so called enrichment information (e.g. RAN data analytics that could be exploited by the near-RT RIC). For this purpose, the non-RT RIC may rely on other RAN OSS functions such as the DCAE, the OOF or the PF.

Complementing the A1 interface, the interaction between the RAN management and orchestration platform and the underlying RAN nodes also rely on the adoption of other standardized interfaces named as O1 and O1* in Fig. 1. In particular, O1 refers to the set of service-based management interfaces being standardized by 3GPP for configuration, performance and fault management of the RAN [8]. In turn, O1* refers to the interfaces developed by ETSI NFV [9] to support the virtualized resource orchestration in a NFVI for the RAN components implemented in a virtualized form.

In addition to the RAN management and orchestration platform, the design and operation of the RAN also relies on

the use of RAN engineering systems such as radio planning and mobile network testing tools. Radio planning tools can provide a detailed modelling of the radio propagation environment (e.g. 3D building maps) and allow operators to conduct estimations of the expected coverage/capacity/quality footprint achievable with potential new sites or technological upgrades of existing ones. On the other hand, radio testing tools (e.g. scanners, drive testing terminals and post-processing platforms) are necessary to test on the field the actual performance of the network and identify potential issues to be corrected in both pre-launch and post-launch network optimization.

The applicability of ML-based techniques is permeating practically all layers of the overall RAN architecture and accompanying RAN OSS for network management and system engineering. Notably, across the RU, DU and CU elements in charge of the radio protocol stack processing, ML techniques can be used for PHY layer processing within RU and DU elements and for MAC scheduler implementation in the DU. Within the near-RT RIC, ML techniques can be used to enhance the operation and support the implementation of the RRM algorithms embedded there. And within the RAN OSS, ML techniques are poised to be central for data analytics information extraction and for decision making logic in operational optimization and automation. The following sections develop each applicability domain.

III. ML FOR PHY LAYER PROCESSING

The PHY layer includes different functions such as:

- Channel coding: At the transmitter side, this function takes as input the bits coming from the MAC layer and generates a new set of bits that include redundancy so that, at the receiver side, this redundancy can be used to detect and correct errors.
- Modulation: This function transforms the coded bits into complex modulation symbols whose amplitude and phase is given by the constellation of the modulation in use. At the receiver side, the demodulation process determines the bits based on the received symbols.
- Layer mapping: At the transmitter side, this function distributes the modulation symbols across the different transmission layers. A transmission layer can be understood as each one of the flows of information that are transmitted on the same time/frequency resources when making use of spatial multiplexing in a Multiple Input Multiple Output (MIMO) system, which refers to having multiple antennas at the transmitter and at the receiver sides.
- Multi-antenna precoding: At the transmitter side, this function determines the symbols to be sent through each antenna port based on the symbols of each transmission layer. This is carried out by applying a precoding matrix.
- Channel estimation and equalization: This function estimates the channel response at the receiver and uses this estimation to counteract the channel effects

when detecting the modulation symbols transmitted in each layer.

- Resource mapping: This function maps the symbols of each antenna port to the set of available resource elements, i.e. the subcarriers of the Orthogonal Frequency Division Multiplexing (OFDM) technique.
- OFDM signal generation and detection: At the transmitter side, this function takes as input the modulated symbols to be transmitted through each subcarrier in each antenna port and executes an inverse FFT to generate the OFDM symbol that aggregates all the subcarriers. This is followed by the cyclic prefix insertion in order to counter-act the effects of multi-path. At the receiver side, the opposite procedures are carried out, i.e. cyclic prefix extraction and FFT to extract the modulation symbols of each subcarrier.
- Beamforming: This function applies a spatial filter in order to form antenna beams that point the energy of the transmitted signal on specific directions.

Within these PHY functions, ML techniques find several applicability areas. One of them is blind detection. Some examples include the use of ML for blind detection of MIMO signals, i.e. without explicitly knowing the channel state information at the receiver, or for blind detection of the modulation used by a received signal [10]. Considering that this type of problems can be usually modelled as deciding one class among multiple classes (e.g. one modulation among a possible set of modulations), classification schemes based on supervised learning become appropriate solutions, relying on e.g. (deep) neural networks, Support Vector Machines (SVM), etc. Such ML-enabled blind techniques facilitate the reduction in the overhead used to communicate the characteristics of the transmitted signals. The real time constraints for the operation of these techniques are very strong, since detection has to be performed at the symbol level, i.e. μs time scale. Also, a key challenge in this case is to specify the adequate dataset for training the classifiers.

ML can also be exploited for channel coding. In this regard, ML techniques can be used to design error correction codes without relying on the expert knowledge on coding theory, which has been the classical approach for code design [11]. The use of Reinforcement Learning (RL) allows iteratively refining the code construction and this allows dynamically responding to changes in the environment where the code is operating.

Channel prediction is another applicability area. This consists in anticipating the deep fading events associated to the channel multipath in wireless communications. For this purpose, ML predictors, e.g. based on recurrent neural networks, can be used. Anticipation of channel conditions can then be used for supporting the dynamic resource allocation, link adaptation or power control.

IV. ML FOR MAC SCHEDULING

The MAC scheduler within the DU is a complex

function that jointly copes with:

- Dynamic resource allocation, in charge of assigning the available resources in the time/frequency/spatial domains to the different users. This might include configurations with multi-carrier scheduling (carrier aggregation, dual connectivity, coordinated MultiPoint) and Multi-User MIMO.
- Link adaptation, in charge of deciding the modulation and coding scheme to be used in each transmission.
- Beam management, which involves different functionalities for acquiring and selecting the appropriate beam for establishing the communication with the mobile terminal, for tracking the beam as the user moves and for detecting and recovering beam failures.

Such complexity turns MAC scheduling into a key applicability area for the use of ML. The practical mobile communications environments are characterized by a multiplicity of random effects, such as mobility, interference, fast fading, traffic generation, etc., which vary at short time scales. Therefore, traditional optimization techniques, which rely on tractable mathematical models of the problem describing the objective function and the feasible set of values of the resource to be allocated, become unpractical because, whenever any of these parameters changes, the optimization would have to be solved again. Instead, the use of ML techniques, e.g. based on Artificial Neural Networks (ANNs), can become more adequate and provide the flexibility to adapt to the different operation conditions [12].

A key requirement when applying ML to the scheduling function is the capability of operating at a very short term time scale. This function is typically executed on a millisecond basis, and may be even shorter for certain service types such as Ultra Reliable and Low Latency Communications (URLLC). Thus, ML-based techniques must be highly responsive to deal with the dynamics associated to the radio environment.

ML can be exploited for the scheduling function from different perspectives. A first approach is to use ML techniques, e.g. supervised deep learning, to dynamically adjust and optimize the operational parameters of the scheduler algorithm (e.g. thresholds, decision-making policy) or even to switch among different scheduling algorithms supported in the DU. The ML algorithms used to drive this adaptation would operate at much longer time scales and can be implemented either as part of the DU or externally (e.g. optimization application executed in the near-RT RIC in case of control loop latencies in the order of seconds or within the RAN management systems for longer control latencies). While this approach can be relatively easy to implement and may fit with the strict time constraints associated to the scheduling operation, the obtained performance would be bounded by the performance limits of the pre-defined algorithms in hand.

To overcome this limitation, another more disruptive approach, as discussed in [13], could be the use of deep RL to eventually figure out how to best tune a neural network to

handle the scheduling problem, just by providing an objective outcome and letting a deep RL-based scheduling algorithm fully embedded in the DU to explore the environment to determine the adequate policy, which will be captured by the internal structure of the neural network. The successful implementation of this type of approach needs to deal with a number of challenges:

- The relevant metrics reflecting the network state as the input of the algorithm need to be properly selected so that they are representative of the actual operation conditions.
- It is essential to make sure that the deep RL model is able to explore all the possible situations of the network and that this exploration, which could lead to sporadically bad decisions, has minimum impact on the network performance. For this purpose, it could be possible to rely on a mix of offline exploration, executed on a network model rather than on the real network, to make a first tuning of the network, and online exploration executed on the actual network.
- It is expected that the obtained algorithm performance will be sensitive to the configuration of the neural network in terms of the number of layers, neurons per layer, etc. Therefore, this sensitivity needs to be assessed and a proper neural network design needs to be devised.
- The learnt scheduler policy will be valid as long as the stationarity conditions where the policy has been learnt do not change. For example, changes in the environment where the DU is operating could lead to different user distributions, new statistical patterns of the interference, etc. Therefore, it is important to ensure the robustness of the algorithm to adapt to this type of changes and re-learn the scheduling policy.

V. ML FOR NEAR-REAL TIME RRM ALGORITHMS IMPLEMENTATION

The RRM algorithms within the near-RT RIC drive different functions such as:

- Radio Bearer (RB) management: Understanding the radio bearers as the constructs that allow transferring data or signaling between the mobile terminal and the CU/DU, RB management addresses the set-up, maintenance and release of RBs, as well as the radio admission control to decide on the acceptance or rejection of a new RB set-up.
- Mobility management: This includes the handover function to decide when to change the serving cell of a connected mobile terminal and the cell (re-)selection function to control how the idle terminals determine their serving cell.
- Interference management: This encompasses functions for detecting and mitigating the interference in a cell.
- RAT and band selection: Given the multiple RATs (e.g. LTE, 5G NR, etc.) and frequency bands (e.g. 700 MHz, 1.8 GHz, millimeter wave frequencies,

etc.) that can be jointly exploited, this function should smartly select the most convenient RAT and band for every mobile terminal in the network.

In contrast to the 1 ms (or shorter) time scale governing the MAC scheduling operation, near-RT RRM algorithms operate under less stringent time constraints given the longer time scales of the system dynamics to cope with (e.g. slow fading channel variations, session-level traffic activity) and the more relaxed execution times of the related control procedures (e.g. activation of a RB, handover execution) with latencies that may typically fall within a 10ms – 1s range [2]. Indeed, the execution of the near-RT RRM algorithms can be either event-triggered (e.g. admission control decision triggered by a RB setup request) or done in a periodic manner (e.g. interference management mechanism recurrently evaluated every few seconds).

In this context, one applicability area of ML techniques comes from the exploitation of prediction in time scales of seconds. To that end, measurements available within the RAN can be used to derive, through proper trained ML algorithms, short-scale forecasts of service request arrivals, user mobility events, path loss variations, etc. and exploit these forecasts to enforce proactive and anticipative RRM decisions [14].

For example, under the scope of mobility management, the handover function can be enhanced by incorporating the capability of predicting trajectories of mobile terminals a few seconds in advance. This allows anticipating future handovers and ensuring the availability of resources at the new cells, thus reducing the dropping rate experienced by the users. Similarly, the possibility of anticipating high interference situations (which could e.g. be associated to user trajectories moving at cell edges), can support the coordination of transmissions between neighbor cells to mitigate the interference experienced by the users.

To carry out the predictions, the recent historical information about the necessary indicators (e.g. recent positions of a user trajectory, recent status of the transmission buffers, recent interference, etc.) should be taken as an input and generate as output the next predicted value of this indicator. For predictions targeting a horizon of few seconds, it is expected that the historical information will span across a time window of a similar order of magnitude. Such information can be stored in the R-NIB of the near-RT RIC for all the users involved in the prediction, and updated dynamically. Possible ML-based prediction techniques to be used include Artificial Neural Networks (ANN) and Support Vector Regression (SVR).

A key aspect in ML-based prediction is to properly train the model (e.g. the ANN weights) so that it can provide accurate results. The training process will typically rely on past observations of the predicted indicator encompassing a sufficiently large number of situations so that the temporal behavior can be adequately captured in the model. As introduced in the RAN architecture description in Section II, capabilities for ML model training and update, along with the necessary functions for deployment of ML models within the RAN nodes, are envisioned to form part of the non-RT RIC within the RAN OSS. In this respect, it is an

open point to analyze whether the same trained model can be used to make predictions from different cells and users, meaning that a single training fed from historical data of multiple cells and users could suffice, or specific training on a per-user and/or per-cell basis needs to be carried out. It is also worth mentioning that, while the prediction process can be associated to the specific RRM algorithm (e.g. handover) and executed at the near-RT RIC, the trained model updates within the non-RT RIC will occur on a longer-term basis. It also deserves attention to analyze whether the resulting trade-off between increased accuracy and required historical information for properly training the models is sufficiently favorable to opt for an ML-based prediction approach in front of simpler schemes (e.g. ARIMA, Holt-Winters, etc.).

Going beyond the exploitation of prediction models to support anticipatory and proactive RRM at the near-RT RIC, it is also possible to use ML techniques as the basis to implement the decision making logic of some of the RRM algorithms. In particular, RL techniques such as Q-learning, would allow decision making policies that maximize a long-term reward function to be automatically learnt based on the dynamic interaction with the environment through an exploration/exploitation process. Besides, the combination of Q-learning with deep learning facilitates an efficient learning in large dimensional problems with large number of states and/or actions [15][16]. Examples of applicability of (deep) RL include, among others, learning the admission control policy when setting up new RBs or power control adjustment for interference mitigation. The operation time scale of these functions, in the order of seconds, enables a relatively fast exploration process for quickly learning an adequate policy. In any case, given the variability of the environment under non-stationary conditions, once a policy has been learnt, it is important to keep some exploratory behavior to let the system adapt to new conditions that may arise in practice.

VI. ML WITHIN THE RAN OSS

The RAN OSS provide all the necessary functionality for deployment, troubleshooting, and management of the RAN. In general terms, these systems operate at longer-term time scales than the functions discussed in previous sections. They could be in the order of minutes, hours or even days, depending on the specific functionality. But given the higher complexity of the next generation RAN compared to previous generations, two capabilities have emerged as central requirements for RAN OSS: *data analytics*, understood as the capability of interpreting and extracting knowledge from the huge amount of raw data generated in the RAN, and *automation*, understood as the capability of minimizing human intervention for the operation of the network. In this regard, as introduced in Section II, the RAN OSS is evolving to embrace a rich set of components (SO, DCAE, OOF, PF, CLAMP, non-RT RIC, etc.) that collectively enable the implementation of different SON functionality covering a whole range of functions (e.g. from self-deployment to self-healing) and able to exploit the huge amount of raw data generated within the RAN. In addition to legacy 4G SON functions

such as Capacity and Coverage Optimization (CCO), Mobility Robustness Optimization (MRO) or Mobility Load Balancing (MLB), some examples of other SON functions that are also under consideration for 5G networks are [17]:

- Service quality optimization, to monitor the performance of specific service quality indicators and adjust the system configuration in case that performance degradation is detected.
- Self-establishment of Network Functions (NFs) including automatic network configuration data handling so that a new-installed NF, such as a CU or a DU node, can be configured in automated manner with no or minimal manual intervention.
- Automatic Network Slice Instance (NSI) creation, in order to enable operators to automatically create NSIs with the necessary NFs and configuration, based on the requirements given by the customers of each slice.

Fig. 2 illustrates the possible implementation of the control loop in the realization of SON functions leveraging data analytics within the RAN OSS. Represented in clockwise direction: the DCAE subsystem collects data from the RAN and produces the relevant data analytics information (e.g. predictions, patterns, anomaly detection) and events; the OOF subsystem relies on the DCAE outputs, together with the applicable constraints and objective functions retrieved as policies from the policy platform, to make decisions on the necessary SON actions; and finally, actuation systems, such as the non-RT RIC or the SO, execute the SON actions in the network. On this basis, Fig. 2 also highlights two main areas of applicability of ML techniques: data analytics information extraction and decision-making in SON functions. A discussion on these two areas of applicability follows.

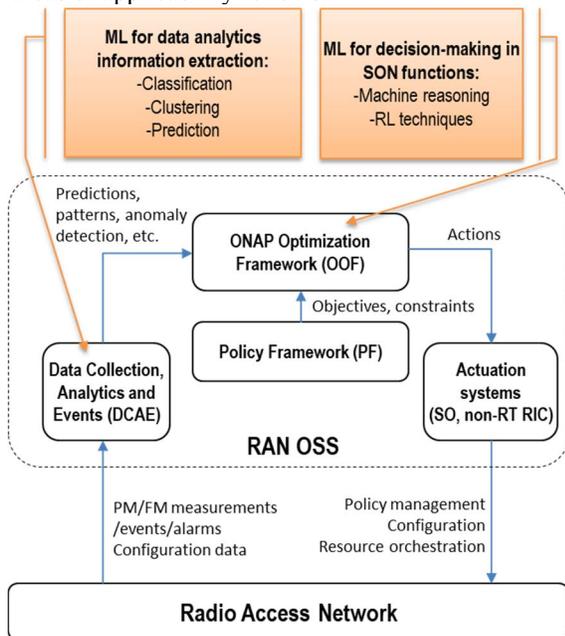


Figure 2. Illustration of the implementation of the control loop for SON functions and the applicability areas of ML techniques within the RAN OSS

A. Data analytics information extraction

Data analytics become fundamental to support almost all of the RAN management areas (capacity planning, performance monitoring, fault management, SON, root cause detection in radio testing measurements) in the complex scenarios where next generation networks will operate. Knowledge gathered via data analytics shall allow better understanding the network behavior under different circumstances and in this way enable smarter decision making mechanisms.

As illustrated in Fig. 2, the processing of the data collected from the RAN (e.g. Performance Management [PM] and Fault Management [FM] measurements, events and alarms) for deriving the necessary insights and models is likely to involve the applicability of different ML techniques for e.g. classification, clustering and prediction. For example, classification can be used to categorize the health of a cell based on collecting performance metrics. In turn, clustering can be used to identify groups of cells having similar behavior in terms of traffic and/or propagation conditions. This would allow exploiting the knowledge learnt for one cell when making decisions for other cells of the same cluster. Similarly, temporal predictions of performance indicators could allow anticipating situations in which the cell may be at risk and take the necessary corrective actions. In the context of radio planning, predictions of future traffic demands on a given geographical area over a long-term time horizon (i.e. weeks, months) could also be exploited for automated planning. An important challenge here relies on the accuracy that can be achieved over large time horizons, given also the fact that traffic demands may be affected by multiple external and uncontrollable aspects (e.g. appearance of new applications in the market, new business players, political and regulatory changes, etc.).

The inherent peculiarities of the wireless mobile environment must be accounted for when developing ML-based data analytics models, in order to ensure their accuracy and their efficient exploitation. Specifically, one of the fundamental assumptions behind many supervised ML algorithms to be used, e.g. for classification, is that the characteristics captured in the data used to train the system will remain the same once the system is deployed. However, this important assumption is often violated in practice, for example, because of the non-stationarity of the environment. Then, if the training data does not follow the same probability distribution as the data over which the trained system is applied, standard supervised learning algorithms will suffer performance degradation. Consequently, robust ML-based data analytics need to incorporate mechanisms to identify when a learnt model is no longer valid and another one needs to be learnt.

Another relevant issue to consider is the problem of imbalanced learning [18], in which an algorithm has to learn from datasets where some situations may be extremely underrepresented. This can be particularly critical for troubleshooting or self-healing solutions when trying to detect situations that rarely occur in the real network, e.g. anomalous cell behaviors, anomalous traffic levels, faults,

etc. and therefore they can hardly be represented in the datasets used to train a ML algorithm for detecting those situations. To handle this problem, ML algorithms need to be devised including specific mechanisms to counteract the imbalance, such as sampling methods that modify the distribution of the imbalanced data set, or cost-sensitive learning that accounts for the cost of erroneously classifying the rare situations.

RAN data analytics information can be exploited by multiple consumers. Within the RAN OSS, different capabilities and tools (e.g. automated control loops, remediation systems, radio planning, RAN KPI testing) can be actually driven by RAN data analytics. Outside the RAN OSS, data analytics information can be consumed by management systems tools above the RAN OSS such as end-to-end service management and business management. And remarkably, the RAN data analytics could also be fed back to the underlying RAN nodes, which can exploit the data analytics services to drive the operation of the RRM algorithms. As an example of this last case, in the area of RB management, the availability within the near-RT RIC of information about service patterns (e.g. statistics and/or predictions of session durations and estimated traffic volume) can help in optimizing admission control decisions by better estimating the resource requirements of the incoming service requests. This can prevent situations in which the cell does not have sufficient resources to serve the traffic demands of the connected users, thus reducing the congestion probability.

B. Decision making in SON functions

RL techniques, capable of progressively learning the actions to apply based on an interaction with the environment, can be used for the implementation of the decision-making logic of some SON capabilities. Indeed, as discussed in [19], RL techniques have been already studied to implement SON functions such as CCO, MRO and MLB. In this case, an important consideration is the trial/error behavior of the RL algorithms when trying different actions during the exploration process. This, together with the long-term operation of the SON functionalities, will mean that, after applying an action (a new configuration), the system will keep it for a long time until generating the reward that reflects how good or bad this action was. Therefore, it may lead to situations in which the network is operating under inadequate conditions for a long period of time, which may severely degrade user experience. In light of the above, the robustness of the RL algorithms to deal with these situations needs to be carefully assessed. The possibility of using off-line exploration based on network models becomes a possible solution to avoid this issue.

In case of complex network operation workflows (e.g. automated remediation systems in large-scale deployments with heterogeneous technologies), machine reasoning techniques emerge as a core technology for achieving automation [20]. Machine reasoning engines for network automation shall be able to leverage human knowledge and expertise to address complex tasks such as root cause analysis, detect issues and vulnerabilities, and either

manually or automatically perform corrective actions. To this end, machine reasoning engines rely on the creation of symbolic models (used to capture concepts and entities, their relations, and behaviors in a machine-processable form) and apply logical techniques such as inference and deduction on non-numerical data like facts, relations and rules [21]. While reasoning engines are starting to be introduced by vendors within their network management solution suites (e.g. guided remediation capabilities for network assurance [22]), its full development and effective integration with data analytics for network automation constitutes an open research area.

VII. CONCLUSIONS

Starting from the architecture for a next generation RAN proposed by Open RAN Alliance, this paper has established a taxonomy and a comprehensive overview of the main areas where machine learning can play a relevant role for the implementation of diverse functions within the RAN nodes as well as within the associated OSS.

At the PHY layer, ML techniques present applicability for performance optimization in areas such as blind detection, channel coding design or channel prediction. In such cases, real time constraints are critical and may range from the microsecond time scale for signal detection to milliseconds for channel prediction. In turn, at the MAC layer, ML can be useful to optimize the dynamic resource allocation. This can be done either by applying ML to optimize the parameters of a given scheduling algorithm or directly by letting a RL learn the optimum policy. Challenges related to the adequate definition of the network state, the performance impact during exploration phase or the robustness in front of non-stationary conditions have been identified.

ML-based prediction tools are envisaged as important components of near-RT RRM algorithms operating in the time scale of seconds and typically associated to user-specific events such as session generation, handover, etc. By properly anticipating the mobility/traffic behavior in the next seconds better decisions can be made, leading to enhanced quality of experience. In this case, it deserves attention to analyze the resulting trade-off between increased accuracy offered by ML models and required historical information, in order to assess the actual benefits with respect to other simpler approaches.

Finally, within the RAN OSS, ML techniques are poised to be central for both data analytics processing and operational automation and optimization. ML-based data analytics enable the extraction of information (e.g. predictions, trained models) that can be exploited by different management functions within the RAN OSS (e.g. SON applications, troubleshooting) as well as from outside (e.g. end-to-end service management, RRM algorithms within RAN nodes). With regard to operational automation and optimization, RL and machine reasoning techniques are clear candidates to be used in the decision-making logic of many SON applications. The robustness in front of non-stationary conditions, the unbalanced learning and the effective integration of machine reasoning engines with data

analytics are key challenges to address in this context.

ACKNOWLEDGEMENT

This work has been supported by the Spanish Research Council and FEDER funds under SONAR 5G grant (ref. TEC2017-82651-R).

REFERENCES

- [1] O-RAN Alliance, "Operator Defined Next Generation RAN Architecture and Interfaces", <https://www.o-ran.org/>, Accessed October, 2019.
- [2] O-RAN Alliance, "O-RAN: Towards an Open and Smart RAN", white paper, October, 2018.
- [3] 3GPP TS 36.300 v15.7.0, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 15)", Sept., 2019
- [4] 3GPP TS 38.300 v15.7.0, "NR and NG-RAN Overall Description; Stage 2 (Release 15)", September, 2019.
- [5] Ericsson, "ONAP and the telecom industry's open-source journey", available at <https://www.ericsson.com/en/white-papers/onap-and-the-telecom-industrys-open-source-journey>, March 2019.
- [6] 3GPP TR 28.890 v16.0.0, "Study on integration of Open Network Automation Platform (ONAP) and 3GPP management for 5G networks", March 2019.
- [7] O-RAN Alliance, "O-RAN A1 interface: General Aspects and Principles Version 1.0", October, 2019.
- [8] 3GPP TS 28.530 v16.0.0, "Management and Orchestration; Concepts, use cases and requirements (Release 16)", September, 2019.
- [9] ETSI GS NFV-MAN 001, "Network Functions Virtualisation (NFV); Management and Orchestration", December, 2014.
- [10] Q. Mao, F. Hu, Q. Hao, "Deep Learning for Intelligent Wireless Networks: A Comprehensive Survey", IEEE Communications Surveys and Tutorials, Vol. 20, No. 4, 2018, pp. 2595-2621.
- [11] L. Huang, H. Zhang, R. Li, Y. Ge, J. Wang, "AI Coding: Learning to Construct Error Correction Codes", January, 2019, <https://arxiv.org/pdf/1901.05719.pdf>
- [12] A. Zappone, M. Di Renzo, M. Debbah, "Wireless Networks Design in the Era of Deep Learning: Model-Based, AI-Based, or Both?", IEEE Transactions on Communications, June, 2019.
- [13] K. Shakil, "AI Radio Access Network", October, 2018, <https://www.linkedin.com/pulse/ai-radio-access-network-kashif-shakil/>
- [14] A. Azari, M. Ozger, C. Cavdar, "Risk-Aware Resource Allocation for URLLC: Challenges and Strategies with Machine Learning", IEEE Communications Magazine, March, 2019, pp. 42-48.
- [15] F. D. Calabrese, L. Wang, E. Ghadimi, G. Peters, L. Hanzo, P. Soldati, "Learning Radio Resource Management in RANs: Framework, Opportunities, and Challenges", IEEE Communications Magazine, September, 2018, pp. 138-148.
- [16] C. Zhang, P. Patras and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," in IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2224-2287, thirdquarter 2019.
- [17] 3GPP TR 28.861 v1.0.0, "Study on the Self-Organizing Networks (SON) for 5G networks (Release 16)", September 2019.
- [18] H. He, E. A. García, "Learning from Imbalanced Data", IEEE Trans. on Knowledge and Data Engineering, Vol. 21, No. 9, September, 2009.
- [19] P. Valente Klaine, M. A. Imran, O. Onireti, R. D. Souza, "A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks", IEEE Comms Surveys & Tutorials, Vol. 19, No. 4, 2017.
- [20] Marin Orlić, Marios Daoutis and Saurabh Singh, "An introduction to machine reasoning in networks", Ericsson Blog, November 2019. Available at <https://www.ericsson.com/en/blog/2019/11/machine-reasoning-networks-introduction>
- [21] Bottou, L., "From machine learning to machine reasoning", Machine Learning, Volume 94, Issue 2, pp 133–149, February, 2014.
- [22] CISCO, "Cisco DNA Assurance User Guide, Release 1.3.1.0", October, 2019.