AI-driven F-RANs: Integrating Decision-making Considering Different Time Granularities

Jonathan M. DeAlmeida, Luiz A. DaSilva, Cristiano B. Both, Celia G. Ralha, Marcelo A. Marotta

***Abstract*—Cloud and fog-based networks are promising paradigms for vehicular and mobile networks. Fog Radio Access Networks (F-RANs), in particular, can offload computation tasks to the network edge and reduce the latency. Artificial Intelligence (AI) techniques can be used in F-RANs to achieve, for example, enhanced energy efficiency and increased throughput. Nonetheless, the appropriate technique selection must consider the different time granularities at which decision-making occurs in F-RANs. We discuss the benefits and challenges of implementing an AI-driven F-RAN considering different timescales, highlighting key Machine Learning (ML) techniques for each granularity. Finally, we discuss the challenges and opportunities to integrate different ML solutions in F-RANs.**

I. Introduction

CLOUD and fog-based networks are promising paradigms to improve vehicular and mobile services via computation offloading. In particular, Fog Radio Access Networks (F-RANs) can reduce the latency by computing tasks at the network edge (*i.e.,* in the fog), closer to the end-users. This feature plays an important role in the context of Artificial Intelligence (AI) advancements and the consequent increase of Machine Learning (ML) applications and services in vehicular and mobile networks.

An F-RAN, depicted in Figure 1, inherits aspects and components from the Cloud Radio Access Network (C-RAN)'s centralized architecture, such as the Base-Band Unit (BBU) pool that processes the workload from geographically distributed Remote Radio Heads (RRHs), which communicate with User Equipments (UEs). Following a fog-computing paradigm, F-RAN distributes core functions, such as computation, storage, communication, and control, extending the processing to the network edge by using Micro Data Centers (MDCs) placed alongside the Radio Access Network (RAN), turning a typical RRH into a Fog-RRH. The F-RAN architecture can be viewed as an evolution from C-RANs, aiming for a balance between centralization and distribution of computational resources [1].

F-RANs can optimize network performance dynamically by taking advantage of processing power near the edge when

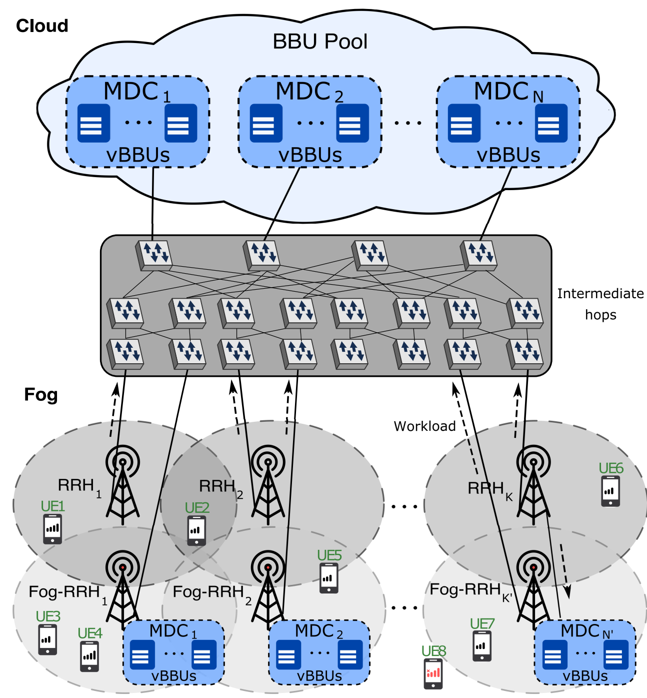

Figure 1. F-RAN system model. The network layer, consisting of RRHs and Fog-RRHs, interfaces with the BBU pool (cloud computing layer) through fronthaul links. The terminal layer, consisting of UEs, together with the network layer form the fog computing layer [1].

available. Dynamic decisions that require up-to-date awareness of network conditions and resource availability result in open challenges, including decision-making regarding edge caching [2], virtual BBU (vBBU) allocation [3], and power consumption minimization [4]. Moreover, F-RANs inherit many challenges that are present in C-RANs, including Central Processing Unit (CPU) scheduling decisions [5], resource block allocation [6], and RRH-UE assignments [7]. The F-RAN's dynamism gives rise to many of these challenges and requires adaptable/smart decisions, rather than approaches that are either hard-coded in hardware or strictly based on utility function maximization. Such resource management decisions usually result in NP-hard problems, typically solved through heuristics and optimization with relaxation techniques before the spread of AI solutions.

ML techniques can find near-optimal solutions for resource management on time, outperforming most heuristics and improving the decision-making process. Several works in the literature propose ML solutions that successfully assist the decision-making task in F-RANs, on timescales ranging from

Jonathan M. DeAlmeida, Celia G. Ralha and Marcelo A. Marotta are with the University of Brasilia, Brazil. (jonathanalmd@gmail.com, ghedini@unb.br, marcelo.marotta@unb.br)
Luiz A. DaSilva is with the Commonwealth Cyber Initiative, at Virginia Tech, United States of America. (ldasilva@vt.edu)
Cristiano B. Both is with University of Vale do Rio dos Sinos, Brazil. (cbboth@unisinos.br)

hours to a few milliseconds. In the scope of hours, online clustering and Neural Networks (NNs) can address decisions regarding the optimal allocation of vBBUs [3]. Within the range of minutes/seconds, service placement, caching, and routing decisions can be made through more sophisticated approaches, such as contextual reinforcement learning [8], deep reinforcement learning agents [9], and adaptive online learning [2]. Decisions made in milliseconds, such as Modulation and Coding Scheme (MCS) prediction for transmission power allocation, resource block scheduling, and CPU scheduling, impose strict time constraints upon AI solutions [4]. It is fundamental to consider the time granularity under which decisions guided by AI need to be made.

In some cases, as in parametric ML, assumptions can simplify the learning process, improving the solution's feasibility but reducing the learning capability. In contrast, non-parametric ML techniques that do not rely on strong assumptions are more flexible, but slower and more data-dependent, to generate the learning model. The latter kind of ML technique can exploit offline training to meet time constraints and improve feasibility. Chien, Lai, and Chao [4] discuss time constraints related to ML approaches, including the difficulty of applying learning techniques to environments with low latency requirements due to the complexity of the training process. In particular, to satisfy the time constraints imposed by different goals, simple learning approaches relying on offline training have been proposed. Nonetheless, the literature that maps potential AI-driven techniques to the timescales in which decisions must occur in an F-RAN is limited, and this article aims to fill the gap.

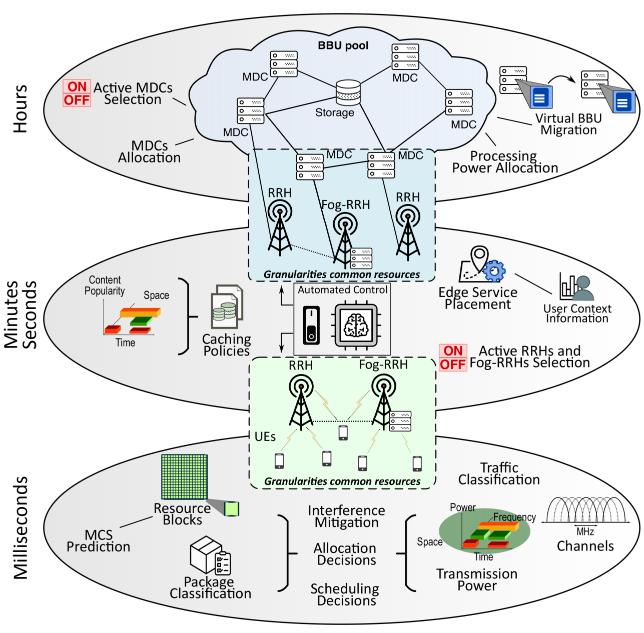

Figure 2. Decision-making in F-RAN, considered at different time granularities.

This article discusses the relationship between ML techniques and decisions associated with different time granularities in F-RANs (depicted in Figure 2). To drive the discussion, we consider the F-RAN architecture proposed in [1] (*i.e.,* an F-RAN is composed of a BBU pool and multiple RRHs/Fog-RRHs) under three granularities: (*i*) hours, (*ii*) minutes/seconds, and (*iii*) milliseconds. Based on a selection of 30 articles (from a systematic investigation of over 100), we examine how ML solutions can improve the performance of F-RANs under different time constraints. We also provide a mapping between (*i*) the time granularity at which resource management decisions must be made in F-RANs and (*ii*) the most suitable ML techniques. The indication of ML techniques for each particular situation is grounded in the literature and considers the time complexity (Big O) of each technique. The readers can refer to Buczak and Guven [10] for an extensive survey and analysis of several ML techniques. Finally, we develop a use case to illustrate the decomposition of the decision-making process into different timescales.

II. Decision-making in Hours

Decision-making that occurs in a timescale of hours incorporates elements from cloud and fog, as depicted in Figure 3. The BBU pool and RRHs/Fog-RRHs are the main components, and decisions associated with such resources have an impact on the F-RAN operation. These decisions result in service migration, a task occurring in a timescale of hours [11]. Considering such a flexible time constraint, it is feasible to apply sophisticated ML techniques with periodic model updating using recent information to improve the RAN performance [3] [4] [12]. We highlight three goals to guide the decision-making in F-RANs, presenting key decisions and the most suited ML techniques.

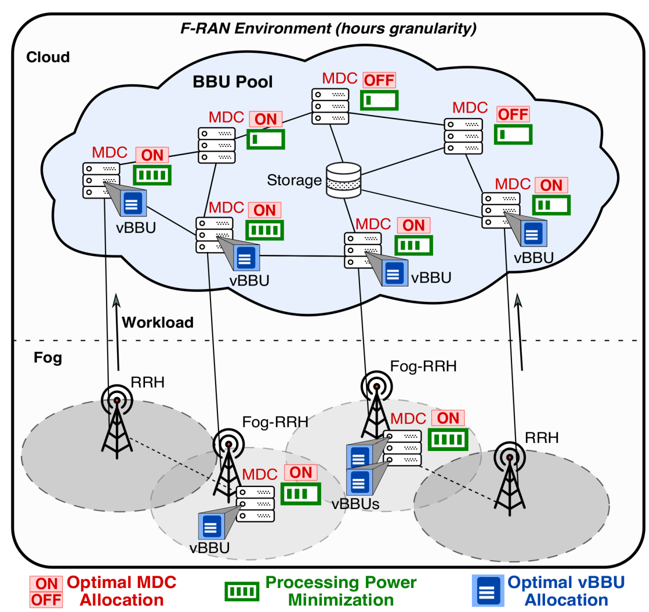

Figure 3. F-RAN: resources and decision-making in hours.

Optimal allocation of MDCs can reduce energy consumption and cost. Decisions consist of selecting the minimal set of active MDCs in the BBU pool that satisfies the F-RAN processing demand. Considering the demand processing dynamicity in F-RANs, accurate prediction of the processing power in use is a key factor in determining the MDCs' future loading. ML techniques capable of calculating regressions during runtime assume an essential role.

Deep recurrent NNs used for performing predictions through regressions of time series is directly applicable to demand prediction. Such techniques cannot process very long sequences and require long training times. Nonetheless, for the allocation of MDCs, in which the time interval for decision-making is large, NNs can be effective. Chien, Lai, and Chao [4] presented a solution applying Long Short-Term Memory, deep recurrent NN, to predict throughput and to allocate MDCs considering a scale of hours.

Minimizing processing power enables MDCs and Fog-RRHs to direct spare resources to serve new F-RANs or to be shared among operators. An F-RAN may serve simultaneously urban, residential, and rural areas that present distinct processing demands. ML techniques able to detect and cluster different demand patterns can benefit from load balancing and processing power allocation. Clustering and NNs techniques can process the data and make decisions that increase the accuracy of workload predictions.

Although NNs can learn a model based on training data and experience, it is important to consider that the high number of parameters increases its complexity and the training process cost grows as the number of layers and perceptrons increases. Nonetheless, this technique is suitable for predicting F-RAN's workload demand. Yu *et al.* [3] presented a solution applying K-means and multi-layer perceptron, examples of clustering and NN, respectively, to detect specific demand patterns of each involved RAN and to minimize the processing power. It is worth mentioning that the optimality of the results obtained with NN and clustering techniques is highly dependent on the quantity and the quality of the input data, *i.e.,* the optimal solution may not be achieved.

Cost reduction of vBBU allocation enables operators to decrease operational expenditures in F-RANs. Decisions must be made to allocate the lowest cost set of vBBUs, by placing them in MDCs that are closer to the edge in Fog-RRHs or in the cloud, to process the RRHs' workload. Market and operational factors introduce processing resource price fluctuations. ML techniques can search the space of solutions to make decisions considering price changes.

Evolutionary computation is suitable for this purpose since it can exploit the search space by generating a population of solutions and combining them to create better-fitted ones. However, the task of setting suitable heuristics and parameters (*e.g.,* population and generation) is not trivial. Compared with solutions based on NNs and clustering, achieving the optimal solution can be challenging since such techniques depend on the algorithm settings. Nevertheless, these limitations are circumvented by applying parameter tuning to empirically evaluate multiple weight and parameter combinations in the algorithm's fitness functions, such as used for vBBU allocation in [12]. Constrained to a timescale of hours, the authors applied a genetic algorithm to improve vBBU allocation while minimizing the operational costs, achieving Pareto optimality, *i.e.,* their solution achieved optimality in the single objective problems from the multi-objective optimization problem.

Although ML techniques have been applied in RANs, there is room to improve the learning process by incorporating the output from finer-grained solutions. Moreover, there is the possibility of improving the solution's performance and feasibility by applying optimization techniques, such as pruning in NN-based solutions and parameter tuning in solutions based on genetic algorithms. Next, we discuss the decision-making from the timescale of minutes/seconds.

III. Decision-making in Minutes/Seconds

Decision-making that occurs in a timescale of minutes/seconds involves resources closer to Fog-RRHs, RRHs and UEs, as depicted in Figure 4. Decisions regarding these resources must consider unexpected circumstances, such as flash-crowd events and content popularity variations. In such dynamic scenarios, a granularity of hours is no longer appropriate for decision-making, and a timescale ranging between minutes and seconds is more suitable. Concomitantly, this dynamicity hinders the application of solutions with costly and offline training processes. It is challenging to keep complex models updated according to recent information due to constant changes in the scenario. Therefore, ML techniques capable of quickly updating the learning model using current information, such as reinforcement and online learning-based solutions, are better suited for dynamic environments with strict time constraints. We focus on three objectives that affect decision-making in F-RANs. For each objective, we present the main decisions and the most suited ML techniques, which are all online, without resorting to offline training.

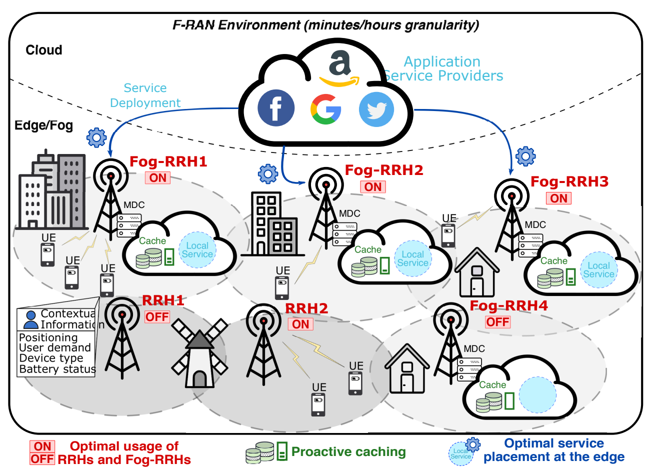

Figure 4. F-RAN: resources and decision-making in minutes/seconds.

Optimal service placement is a crucial goal for Application Service Providers (ASPs) to reduce service deployment costs and improve users' long-term satisfaction. An ASP must select the optimal set of Fog-RRHs to deploy services at the edge considering its limited budget. However, Fog-RRHs' resource availability is constantly in flux and, concomitantly, UEs' contextual information varies significantly (*e.g.,* connected RRH/Fog-RRH and content demand). Consequently, ASPs have to adapt their decisions to these changes. ML techniques that can cross-reference contextual data, resource availability, and ASPs' interests, generating and updating their model in real-time, are well suited.

Contextual reinforcement learning algorithms are employed in recommendation systems to process different sources to maximize the average reward of a given objective. Such techniques can detect errors and make corrections during the learning process, which makes them highly suitable for environments where the data is gathered continuously. A potential challenge is to set a proper reward function for the algorithm, which requires constant computation and data acquisition. As for service placement, data is continuously processed and collected in real-time, which makes this technique suitable. Chen *et al.* [8] implemented a solution, in the timescale of minutes, using bandit learning, a contextual reinforcement algorithm, based on user feedback, for optimal selection of Fog-RRHs.

Enhancing the caching hit rate in F-RANs can improve network latency and content acquisition time. A particular challenge is to decide whether to cache content, via prediction of its popularity and analysis of users' historical requests and consequent update of caching policy in use. Content popularity varies in space and time, so an accurate prediction must consider these variables. Depending on the level of disaggregation of the service, caching decisions range from seconds to several minutes. ML techniques that learn and make predictions in real-time are appropriate for improving the cache hit rate by updating the F-RAN's caching policy.

Online learning algorithms generate sparser solutions, and the model can be updated during runtime, making them highly suitable for large-scale learning tasks with real data. Compared with offline approaches, online learning has a valuable role in dynamic contexts because it can track environmental changes in real-time. However, it is not a simple task to evaluate the model, and it is not trivial to achieve the correct behavior automatically. In this sense, it is necessary to assess the prediction error bounds, referred to as a mistake-bound model. If correctly evaluated, online learning algorithms can be effective in predicting content popularity. Jiang *et al.* [2] present an asymptotically optimal performance by employing Follow-The-Regularized-Leader, an adaptive online learning algorithm, to predict and track local popularity in real-time, and to update the caching policy within minutes, improving the caching hit rate. The authors provide extensive performance evaluation considering bounds for the popularity prediction error, cache hit rate, and caching policy.

Optimal usage of RRHs/Fog-RRHs refers to the selection of the minimal set of active RRHs to serve UEs, meeting their throughput demand. UEs' requests vary significantly within a time interval of minutes or less. A solution able to adapt promptly according to this variation is necessary. For example, reinforcement learning techniques are applied in many domains in which a system interacts with a dynamic environment and learns in real-time.

Deep reinforcement learning can scale highly complex problems by automatically reducing and tuning features directly from runtime inputs. Such a technique can estimate the possible states instead of computing every solution, and the solution space is pared down during the decision process. Also, note that this technique requires a lot of data and computation to achieve effective results. Fortunately, a plethora of data is continuously gathered from RRHs/Fog-RRHs in F-RAN. Xu *et al.* [9] show that a deep reinforcement learning agent can be effectively applied to select the minimal set of active RRHs consonant with user requests and varying scenario dynamics.

Next, we discuss objectives and ML techniques considering the timescale of milliseconds.

IV. Decision-making in Milliseconds

Decision-making that occurs in milliseconds is particularly important for the physical layer, involving spectrum and processing resources, as depicted in Figure 5. Decisions regarding such low level resources have stringent time constraints and must consider the high variability of channel conditions caused by interference, noise, and UE mobility. ML techniques can be exploited to assist decisions constrained to milliseconds [5] [6] [7]. In this case, the adoption of offline training and hardware implementation is key. We highlight three goals for decision-making, presenting key decisions and the most suited ML techniques for each case.

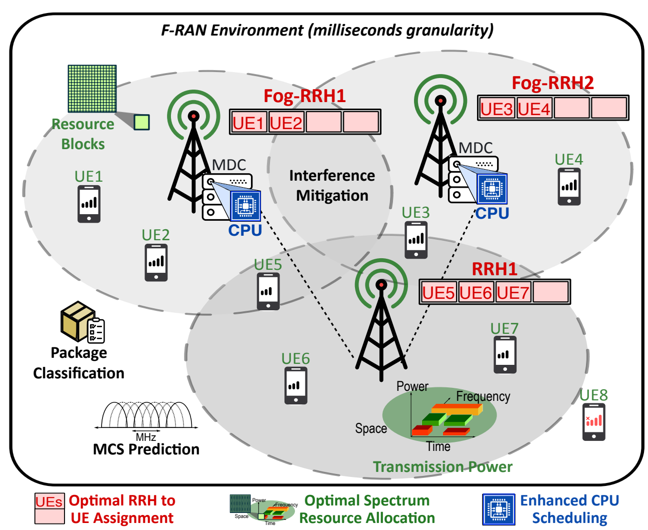

Figure 5. F-RAN: resources and decision-making in milliseconds.

Optimal spectrum resource allocation impacts the F-RANs' energy consumption and spectral efficiency. The decision is made concerning resource blocks and transmission power allocated for communications between RRHs and UEs, considering Quality of Service (QoS) requirements. Since

F-RANs employ spectrum reuse in a dense scenario, appropriate resource block and transmission power allocation are needed to avoid inter-tier interference between RRHs/Fog-RRHs. An ML technique capable of acquiring current state information to quickly assign the available resource blocks and adjust the transmission power to minimize interference is required.

Quasi-statistical model-free reinforcement learning algorithms are used to handle problems with stochastic transitions and rewards. A model-free algorithm that can learn without using the current policy nor accurate representation of the environment is suitable for problems where the environment model is not available. Additionally, compared with other learning techniques, this type of algorithm is considerably less complex in computation and space, enabling its usage in an online manner. Note that previous experience is required to perform the training task since there is no defined model. Alqerm and Shihada [6] show that a model-free learning approach with a Q-learning agent is suitable to perform joint allocation of resource blocks and transmission power while mitigating interference and maintaining QoS even with a strict time constraint.

Enhanced CPU scheduling brings throughput improvements, while tailoring the processing resources to UEs' QoS requirements. Decisions regarding Fog-RRHs' CPUs scheduling may consider the application's delay budget, packet length, and inter-arrival time. AI classifiers that train a model beforehand without online updates are suitable to distinguish between different types of traffic, identifying their patterns in milliseconds. It is worth mentioning that the internal CPU operation (*e.g.,* cache coherence protocols and acceleration features) depends on its specific design/architecture. From the perspective of scheduling algorithm development, a higher-level context, such specific characteristics can be abstracted or included as parameters.

Support Vector Machines (SVM) are effective in high-dimensional spaces and highly suitable for classification using unstructured/semi-structured data. Despite its successful application in many domains, it is worth mentioning that this technique is not suitable for large and noisy datasets, usually requiring hyperparameter tuning with grid search and cross-validation to correctly set the kernel function/parameters. However, for the task of discriminating between different types of traffic, this technique shows great promise. Wang *et al.* [5] report throughput improvements by applying SVM to classify the traffic and adapt the CPU scheduling decisions based on the traffic classification and its QoS requirements, and the baseband signal processing load.

Optimal RRH-UE assignments can mitigate the F-RAN's signaling overhead, reducing transmitted bits to perform a more robust communication in the presence of interference/noise. Decisions consider the best set of assignments among RRHs and UEs to perform communication without interfering with parallel transmissions. F-RANs are employed in dense scenarios with high interference, resulting in frequent channel state changes between RRHs and UEs and requiring frequent Channel Quality Indicator (CQI) transmissions. Considering that the primary use of CQIs is to determine the appropriate MCS, ML techniques bring the opportunity to decide RRH-UE assignments while predicting the appropriate MCS. AI classifiers with an offline trained model can quickly predict MCS based on knowledge about UEs' positions and past transmission beam to decide RRHs-UEs assignments, reducing signaling overhead.

The random forests algorithm can create models based on a few samples and deals well with missing data and unbalanced datasets. Since the algorithm tends to overfit, it is recommended to perform hyperparameter tuning to estimate the performance (*e.g.,* k-fold cross-validation), then select the parameters that minimize the prediction error. Imtiaz *et al.* [7] proposed a solution based on random forests with offline training to improve resource allocation based on MCS prediction using UEs' information.

The improvements that ML techniques bring to F-RANs' operation are clear. However, additional benefits can accrue from sharing information collected at different time granularities. We discuss such integration and trade-offs next.

V. Analysis of ML Techniques in F-RANs

We summarize the ML techniques applied in F-RANs on each timescale and discuss the opportunities, challenges, and trade-offs on integrating ML-based solutions among granularities, including a use case to illustrate the importance of the timescale decomposition.

A. ML Techniques

The discussion above results from an extensive investigation of the literature, starting from over 100 articles, filtered down to 30, which specifically dealt with automated solutions for resource management in RANs. From those, we selected the three most relevant works for each time granularity. Table I presents a mapping between decision-making, granularity, and suitable ML techniques to RAN operation, including the advantages/disadvantages of each solution.

It is worth mentioning that the proper ML technique selection will depend on the context of the application and the data used in the decision-making process. Most of the techniques applied in the hours granularity are variations of NNs, which are characterized by their complex training process. In a granularity of minutes/seconds, online/reinforcement learning-based algorithms play an essential role. Updating the learning model collecting data in real-time is a critical factor for the decisions in this timescale. Regarding the granularity of milliseconds, the main limitation is that training models in real-time are not practical. In this context, solutions overcome this limitation by applying quasi-statistical techniques or previously trained models to classify well-defined categories.

Some techniques are suitable in a range of timescales. SVM and random forests techniques used as classifiers can be applied in solutions from all timescales. It is possible to improve

Table I
MAPPING OF REFERENCES, DECISIONS, TIME GRANULARITY, ML TECHNIQUES, MAIN BENEFITS, ADVANTAGES, AND DISADVANTAGES

Reference	Decisions	Granularity	ML Techniques	Benefits	Advantages	Disadvantages
Chien, Lai, and Chao [4]	Optimal allocation of MDCs	Hours	Long-short Term Memory	Energy consumption minimization and cost reduction	Can remember the information through time, which turns it a lot suitable for time series prediction	Can not process very long sequences; training process is complex (it requires a long training time)
Yu *et al.* [3]	Cloud workload prediction for optimal vBBU allocation	Hours	Multi-layer Perceptron and K-means	Processing power minimization in the F-RAN	Highly suitable for regression tasks; able to learn how to act based on the training data and experience	The number of parameters grows quickly; training process is complex
Aryal and Altmann [12]	Allocate the set of vBBUs at the lowest cost for the network operator	Hours	Genetic algorithm	Reduced F-RAN's operational expenditures	Can generate many different solutions in considerably short time of computation, enabling fast adaptation to the environment changes	Difficult to set a proper heuristic and parameter values; it do not reach the optimal solution at all times
Chen *et al.* [8]	Select the optimal set of Fog-RRHs to be rented to deploy services at the edge of the network	Minutes	Bandit Learning	Optimal service placement and deployment costs reduction while attending QoS demand	Can detect errors and make corrections during the training process; suitable for environments where the data is constantly gathered	It is not trivial to set a proper reward function, requiring a well defined environment; requires constant computation and data acquisition
Jiang *et al.* [2]	Decide whether or not to cache specific content via prediction of its popularity in Fog-RRHs	Minutes	Follow the (proximally) Regularized Leader (Adaptive Online Learning)	Enhanced caching hit rate	Generates sparse solutions; converges fast; the model can be efficiently updated during runtime	It is difficult to evaluate; it is not trivial to achieve the correct behaviour
Xu *et al.* [9]	Select minimal set of active RRHs/Fog-RRHs that meets UE requests in terms of throughput	Seconds	Deep Reinforcement Learning	Energy consumption minimization	Can reduce the complexity of the solution, enabling to scale complex problems; features are automatically deduced and optimally tuned	Requires a lot of data and computation to overcome other techniques (even more than pure reinforcement learning algorithms)
AlQerm and Shihada [6]	Resource blocks and transmission power allocation	Milliseconds	Q-learning	Enhanced spectral efficiency while mitigating interference and maintaining QoS	Can learn without an accurate representation of the environment; less complex in computation/space	Experience is required for training task; it is not able to identify how the dynamics of the environment affects the system
Wang *et al.* [5]	CPU scheduling based on the traffic classification	Milliseconds	SVM	Throughput improvements while meeting UE' QoS requirements	Effective in high dimensional spaces; suitable for unstructured and semi-structured data	Not suitable for large and noisy datasets; hard to select the kernel function and tune the parameters
Imtiaz *et al.* [7]	Select the best set of assignments among RRHs and UEs to perform communication via MCS prediction	Milliseconds	Random Forests	Better spectrum resources allocation; CQI signaling overhead reduction	Can create an effective model with few samples; deal with missing data and unbalanced datasets	Limited control on what the model does; proclivity towards over-fit; complex in space

Q-learning by introducing NNs to approximate the Q-value (deep Q-learning), a well-suited approach for decision-making for both hour and minutes/seconds granularities. Lastly, AI play an essential role since the resource management problem's complexity increases as more performance requirements from different granularities are introduced in the objective function (*e.g.,* latency, throughput, energy consumption, cost). Concomitantly, characteristics of components from short-time granularities (*e.g.,* CPU and RRHs specifications) are important parameters for solutions from longer timescales. Although several works have proposed valuable ML solutions in RANs, there are still opportunities to integrate different solutions.

B. Integrating Multi-objective Decision-making in F-RANs

In integrating solutions from different granularities, the combination of distinct goals results in trade-offs. Divergent objectives involved in decision-making in F-RANs are not mutually exclusive and can be employed together according to the interests of network operators. F-RANs can adopt a flexible decision-making architecture capable of selecting and tailoring the ML techniques according to the highest priority objectives. A multiagent system can potentially be exploited to enable the integration of different ML techniques in the same framework.

Considering the granularity of hours, achieving optimal allocation of MDCs/vBBUs and processing power minimization, some agents can implement NN or evolutionary algorithms. In contrast, others can apply meta-learning algorithms to decide which learning technique to use according to the current scenario, historical data, and the operators' interests. However, different goals that may conflict regarding vBBUs allocation, distance, cost, and the number of cores in use, create trade-offs, such as investigated in our previous work [13]. Some of these trade-offs are depicted in Figure 6. To avoid such conflicts, agents must communicate with each other to make a shared decision comprising vBBU allocation and migration operations guided by the operator's business model. The business model can be represented as a utility function (weighted objective sum) to balance the operator's interests, aiming for equilibrium at the intersection of the two curves in Figure 6.

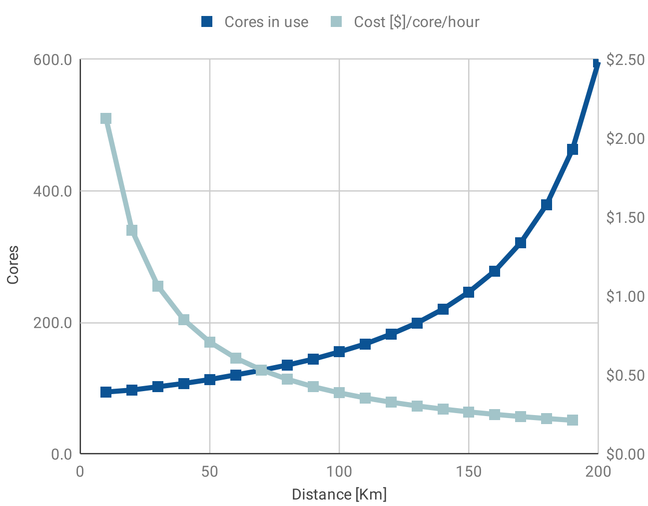

Figure 6. The influence of distance between MDC and RRH on the minimal number of processing cores allocated and the processing cost per hour, illustrating an important trade-off for F-RANs. These results consider the processing of seven iterations of a Forward Error Correction (FEC) decoder [14], an RRH with fixed upstream data rate of 50Mb/s (following the Fujitsu's eNodeB maximum transmission rate) [15], an MDC comprised of processors with characteristics of an Amazon's C5 instance (with an efficiency of 8 operations per cycle and 3.4 GHz per core, and allocation price of $0.0425/hour/core).

In our previous work [15], based on Marotta *et al.* [13], we propose an optimal solution for vBBU allocation in F-RANs by deciding RRHs-MDCs assignments for cost minimization, considering the trade-off depicted in Figure 6. The optimization problem formulation minimizes the sum of the system's processing power allocation cost for each vBBU class within each MDC, computing RRHs' workloads per time slot t. This minimization is subject to three constraints.

1) The horizontal allocation constraint, *i.e.,* an MDC must process all the assigned RRH's workload in parallel considering the available computational power (*e.g.,* number of cores).
2) The vertical allocation constraint, *i.e.,* the execution time for an MDC's processor, considering their maximum clock to compute the largest non-parallelizable part of the wireless workload (*e.g.,* decoding a code-block), must meet delay restrictions to make an MDC-RRH assignment.
3) The assignment constraint is required to assure the assignment of all RRHs' workloads to at least one MDC, ensuring that all the RAN's workload is going to be processed.

For a more rigorous and formal description of the problem formulation, please refer to De Almeida *et al.* [15]. Here, we focus on demonstrating the impact of decomposing timescales.

The solution was evaluated under downtown/urban and suburban scenarios using real demand data provided by an operator. In these scenarios, we assess the effect of the selection of timescale for the decision making by applying the optimal solution and analyzing the number of migrations per MDC per day (Figure 7). The number of migrations increases as the time interval for decision-making decreases. Considering a sample of 35 and the application of a t-test with a 95% confidence interval, the average number of migrations per MDC per day for a time interval of 10, 30, and 60 minutes are 168.4±4.8, 65.7±1.7, and 36.3±1.0, respectively. The number of migrations for a time interval of 10 minutes and 30 minutes is approximately 4.6 and 1.8 times higher than for a time interval of one hour. Analyzing the spatial aspect, the number of migrations is higher in urban areas, in which Internet traffic is intense and fluctuates more. It is crucial to decide a proper time interval for decision-making since each migration has an associated cost depending on the migration time and transmission cost [11]. Moreover, a longer timescale can enable the application of more sophisticated ML techniques and frequent updates in the model.

At the granularity of minutes/seconds, the varying behavior of F-RANs hinders the usage of techniques based on recurrent offline training since the information cannot be well represented by historical data. Even with this limitation, there is the opportunity to integrate different ML techniques at this granularity. The objectives can be divided among agents implementing different ML algorithms. Some agents implement reinforcement learning-based algorithms for optimal usage of RRHs, while others implement adaptive learning algorithms for content popularity prediction to enhance caching hit rate. Nonetheless, these different goals may conflict. The algorithm might decide to turn off one or more RRHs that are near MDCs, which were selected to deploy the ASP's local services for their users. Communications between both agents enable system adaptation to arrive at a shared decision according to operators' interests, reaching a trade-off between decreasing energy consumption costs or increasing revenue through

MDCs service hosting. A contract may be settled during runtime by adjusting the price for service deployment to cover the energy costs related to sustaining an RRH turned on.

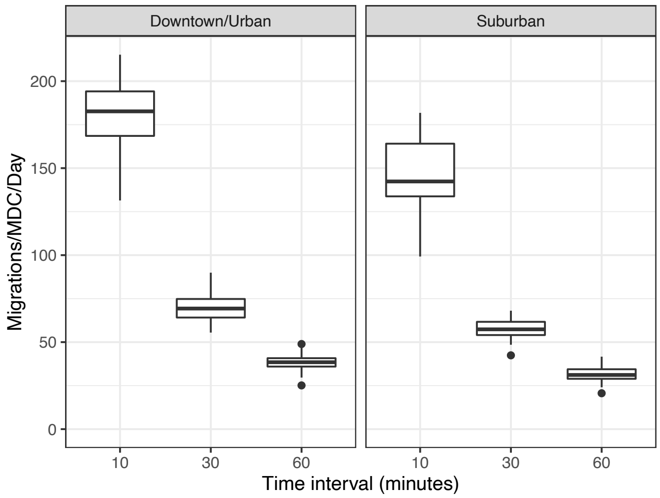

Figure 7. Number of migrations per MDC per day in Milan (Italy), considering the trade-off between processing power and distance between MDC and RRH evaluated in different time granularities under three time intervals: (*i*) 10 minutes; (*ii*) 30 minutes; and (*iii*) 60 minutes. The use case is based on data from five weeks (sample size equals to 35). The different regions (downtown/urban and suburban) were defined based on [15], using unsupervised learning (k-means clustering). Internet traffic input data obtained from Telecom Italia.

It is essential to highlight that some AI classifiers can be applied at milliseconds' granularity because the training task is performed beforehand. The recent data does not have a considerable impact on the classifier, and an effective model can be achieved using historical data since the predictions are limited to distinct categories (*e.g.,* traffic type and MCS indexes). There is the opportunity to adopt a multiagent-based solution. Two agents with the same goal (*e.g.,* traffic classification) can achieve it by applying different algorithms, such as boosted trees and naive Bayes. The decisions regarding RRH-UE assignments and spectral resources share a trade-off in common regarding resource block allocation, enhanced channel capacity, and the number of served UEs. Spectrum is a limited resource, sometimes requiring decisions of whether to serve a new UE or to provide enhanced channel capacity to already connected ones. At least one agent must receive information from the other to reach a shared decision regarding the resource block assignment and UEs being served. The shared decision must be ranked according to a performance objective (*e.g.,* highest overall throughput or fairest scheduling), using a shared database to readjust its objectives aiming for the best outcome.

VI. Concluding Remarks and Future Work

We discussed the benefits and challenges of implementing AI-driven F-RANs. To drive the discussion, we considered ML techniques applied in F-RANs in three time granularities. Considering the characterized granularities, we discussed the challenges and opportunities of developing AI-driven F-RANs. There is still the opportunity to improve the learning process by integrating solutions. As future work, we aim to integrate different decision-making processes across the granularities by implementing one multiagent architecture.

Acknowledgments

This work was conducted with partial financial support from the National Council for Scientific and Technological Development (CNPq), Brazil, under grant numbers 311301/2018-5 and 130555/2019-3. This work was also partially supported by the Commonwealth Cyber Initiative (CCI).

References

[1] M. Peng *et al.*, "Fog-computing-based radio access networks: issues and challenges", *IEEE Network*, vol. 30, pp. 46–53, July 2016.

[2] Y. Jiang *et al.*, "User preference learning based edge caching for fog radio access network", *IEEE Transactions on Communications*, vol. 67, pp. 1268–1283, November 2018.

[3] Y. Yu *et al.*, "Integrating Clustering and Learning for Improved Workload Prediction in the Cloud", in *IEEE 9th International Conference on Cloud Computing (CLOUD)*, pp. 876–879, June 2016.

[4] W. Chien, C. Lai, and H. Chao, "Dynamic resource prediction and allocation in c-ran with edge artificial intelligence", *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 4306–4314, April 2019.

[5] K. Wang *et al.*, "Computing aware scheduling in mobile edge computing system", *Wireless Networks*, pp. 1–17, January 2019.

[6] I. Alqerm and B. Shihada, "Sophisticated online learning scheme for green resource allocation in 5G heterogeneous cloud radio access networks", *IEEE Transactions on Mobile Computing*, vol. 17, pp. 2423–2437, January 2018.

[7] S. Imtiaz *et al.*, "Random forests resource allocation for 5g systems: Performance and robustness study", in *IEEE Wireless Communications and Networking Conference Workshops*, pp. 326–331, April 2018.

[8] L. Chen *et al.*, "Spatio temporal edge service placement: A bandit learning approach", *IEEE Transactions on Wireless Communications*, vol. 17, pp. 8388–8401, December 2018.

[9] Z. Xu *et al.*, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs", in *IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2017.

[10] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection", *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[11] J. Liu, Q. Yang, G. Simon, and W. Cui, "Migration-based dynamic and practical virtual streaming agent placement for mobile adaptive live streaming", *IEEE Transactions on Network and Service Management*, vol. 15, pp. 503–515, June 2018.

[12] R. G. Aryal and J. Altmann, "Dynamic application deployment in federations of clouds and edge resources using a multiobjective optimization ai algorithm", in *International Conference on Fog and Mobile Edge Computing*, pp. 147–154, April 2018.

[13] M. A. Marotta *et al.*, "Characterizing the relation between processing power and distance between BBU and RRH in a cloud RAN", *IEEE Wireless Communications Letters*, vol. 7, pp. 472–475, June 2018.

[14] S. Bhaumik *et al.*, "CloudIQ: a framework for processing base stations in a data center", in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, pp. 125–136, 2012.

[15] J. M. De Almeida *et al.*, "Optimal allocation of vBBUs considering distance between MDC and RRH in F-RANs", in *IEEE International Conference on Communications (ICC): SAC Cloud & Fog Computing, Networking, and Storage Track*, June 2020.

Biographies

Jonathan M. DeAlmeida (jonathan@aluno.unb.br) holds a M.Sc. and a B.Sc. in Computer Science from the University of Brasilia, Brazil. He is currently pursuing a B.Sc. in Statistics at University of Brasilia. He is a prospective Ph.D. student at Virginia Tech, United States of America. His research involves Radio Access Networks, Next Generation Networks, Cognitive Radio Networks, Artificial Intelligence, and Machine Learning.

Luiz A. DaSilva [F] (ldasilva@vt.edu) is the Executive Director of the Commonwealth Cyber Initiative and the Bradley Professor of Cybersecurity at Virginia Tech. He was previously at Trinity College Dublin, where he was the director of CONNECT, the Science Foundation Ireland Research Centre for Future Networks. He is an IEEE Fellow, and an IEEE Communications Society Distinguished Lecturer.

Cristiano B. Both (cbboth@unisinos.br) is an associate professor of the Applied Computing Graduate Program at the University of Vale do Rio dos Sinos (UNISINOS), Brazil. He has coordinated research projects funded by H2020 EU-Brazil, CNPq, FAPERGS, and RNP. His research focuses on wireless networks, next generation networks, softwarization and virtualization technologies for telecommunication network, and SDN-like solutions for the Internet of Things. He is participating in several Technical Programme and Organizing Committees for major international conferences.

Celia G. Ralha (ghedini@unb.br) holds a Ph.D. from Leeds University, England and a M.Sc. in Electronic and Computer Engineering from Aeronautics Institute of Technology, Brazil. She is an associate professor of the Informatics Graduate Program, University of Brasilia, Brazil. She receives a research productivity grant from the Brazilian National Council for Scientific and Technological Development (CNPq). Her current research interests include intelligent systems, multi-agent systems, agent-based simulation, multi-agent planning and cognitive networks.

Marcelo A. Marotta (marcelo.marotta@unb.br) is an adjunct professor of the Department of Computer Science of the University of Brasilia, Brazil. He holds a Ph.D. in Computer Sciences from the Institute of Informatics of the Federal University of Rio Grande do Sul, Brazil. His research involves Cloud Radio Access Networks, Wireless Networks, Next Generation Networks, Internet of Things, Software Defined Radio, and Cognitive Radio Networks.