



# An Experimental Assessment of Channel Selection in Cognitive Radio Networks

Anna Umbert<sup>1</sup>(✉), Oriol Sallent<sup>1</sup>, Jordi Pérez-Romero<sup>1</sup>, Juan Sánchez-González<sup>1</sup>,  
Diarmuid Collins<sup>2</sup>, and Maicon Kist<sup>3</sup>

<sup>1</sup> Universitat Politècnica de Catalunya (UPC), c/Jordi Girona, 1-3,  
08034 Barcelona, Spain  
annau@tsc.upc.edu

<sup>2</sup> CONNECT Centre for Future Networks, Trinity College Dublin (TCD), Dublin, Ireland

<sup>3</sup> Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil

**Abstract.** The management of future networks is expected to fully exploit cognitive capabilities that embrace knowledge and intelligence, increasing the degree of automation, making the network more self-autonomous and enabling a personalized user experience. In this context, this paper presents the use of knowledge-based capabilities through a specific lab experiment focused on the Channel Selection functionality for Cognitive Radio Networks (CRN). The selection is based on a supervised classification that allows estimating the number of interfering sources existing in a given frequency channel. Four different classifiers are considered, namely decision tree, neural network, naive Bayes and Support Vector Machine (SVM). Additionally, a comparison against other channel selection strategies using Q-learning and game theory has also been performed. Results obtained in an illustrative and realistic test scenario have revealed that all the strategies allow identifying an optimum solution. However, the time to converge to this solution can be up to 27 times higher according to the algorithm selected.

**Keywords:** Channel selection · Classification · Cognitive Radio

## 1 Introduction

The increasing traffic demand will lead future wireless networks to face a severe shortage of spectrum, especially when considering the highly dense deployments of small cells envisaged for meeting the demands of future systems. Cognitive Radio Networks (CRN), based on the Cognitive Radio (CR) paradigm [1], will bring light to this problem. Briefly, CR observes the environment, analyzes these observations, makes decisions to intelligently configure certain radio parameters, and finally executes these decisions. Analysis and decision can be supported by means of learning mechanisms that exploit the knowledge obtained from the execution of prior decisions.

CRN concepts are also expected to play a relevant role in the context of future 5G (5th Generation) networks [2], which should include by design unprecedented network flexibility and highly efficient/adaptive network resource usage, including flexible

spectrum management. Thus, the introduction of intelligence in the network will be an important requirement. In this direction, the advent of big data analytics [3] will boost the extraction of the meaningful information from the available data, to support the use of cognitive capabilities both in the Radio Access Network (RAN) and in the Core Network.

Using knowledge-based procedures and Artificial Intelligence (AI) as key elements of cognition for supporting the optimization in future networks has been considered in the literature for the last several years. Specific algorithms for learning time domain traffic patterns and mobility patterns, respectively, have been proposed and analyzed [4, 5]. Similarly, in [6] a clustering strategy was proposed to identify the user's daily motifs and extract the personalised Quality of Service observed by a user when being connected to a real 3G/4G network. Nevertheless, the authors believe that one important reason for the (relatively) low penetration of AI concepts in this domain so far is due to the difficulty for the research community in general to test (and hopefully prove the validity of) potential solutions in realistic conditions. Clearly, AI-based knowledge discovery models (e.g. classification, prediction, clustering) can hardly be properly assessed in simulated environments, where many of the real-world effects are not retained. Instead, more solid results and conclusions can be derived from implementing such mechanisms in realistic conditions.

In this respect, WiSHFUL is a European project from the European Horizon 2020 Programme that focuses on speeding up the development and testing cycles of wireless solutions and, therefore, it offers a great opportunity to gain access to realistic data and measurements [7]. It defines software modules with unified interfaces that permit wireless developers to quickly implement and validate advanced wireless network solutions. The WiSHFUL project offers access to different advanced wireless testbeds, among them the IRIS testbed at Trinity College Dublin [8].

In this context, this paper describes a specific experiment using the IRIS testbed. The experiment focuses on the Channel Selection functionality for CRN, so that an access point decides the most appropriate channel to use within a band that is shared among multiple transmitters. This selection is based on a supervised classification that allows estimating the number of interfering sources existing in a given frequency channel. Specifically, four different classifiers have been implemented: decision tree, neural network, naive Bayes and Support Vector Machine (SVM). Additionally, a comparison against other channel selection strategies using Q-learning and game theory has also been performed. In this way, this experiment contributes to expand the capabilities of the existing WiSHFUL Intelligence framework [9] that offers an experimentation environment for early implementation and validation of end-to-end 5G solutions that improve resource utilization through advanced reconfigurability of radio and network settings.

The rest of the paper is organised as follows. Section 2 presents the IRIS testbed used for executing experiments. Section 3 discusses the considered approaches for channel selection. The experimental results obtained with these approaches are presented in Sect. 4, while Sect. 5 summarizes the main conclusions.

## 2 The IRIS Testbed

The IRIS testbed is the reconfigurable radio testbed at Trinity College Dublin [8]. It provides access to radio hardware that supports the experimental investigation of the interplay between radio capabilities and networks.

The testbed employs 18 ceiling or wall mounted Universal Software Radio Peripheral (USRP) N210s equipped with SBX daughterboard, reaching frequencies between 40 MHz and 4.4 GHz, and 4 other radio nodes not available within the WiSHFUL context, as underlying radio resources. All these elements are connected to a private computational cloud, allowing to deploy an array of computational environments. By default, each USRP device of the testbed is associated to a Virtual Machine (VM) that occupies 4 CPU cores and 4 GB of RAM from the computational cloud. Testbed access was supported by jFed Experimenter suite developed by the Fed4FIRE+ EU project.

For setting up and executing the experiments with the IRIS testbed, we modify the code and the configuration files in a remote local machine at Universitat Politècnica de Catalunya (UPC) premises, and then we upload the files to the testbed machines, we execute the test, and we download back the results files to our local machines. To perform these operations, a custom made code implemented with Python programming language that uses the WiSHFUL software framework and the Unified Programming Interface (UPI) functions and runs on the IRIS Testbed has been created.

Two different pieces of python code, namely the *wishful\_controller* and the *agent*, have been used. The *wishful\_controller* runs on a computer, whereas one *agent* runs on each radio node. The configuration of a radio node as a transmitter or receiver is made by the *wishful\_controller* when the radio program is activated. The purpose of the *agent* is to connect to the *wishful\_controller* and wait for instructions (passed through UPI calls). In turn, the *wishful\_controller* executes the logic for controlling the experiment.

A deployment example of the experimentation framework is illustrated in Fig. 1. In this case, a scenario with three nodes acting as transmitters (AP1, AP2, and AP3) and three nodes acting as receivers (STA1, STA2 and STA3) is considered.

## 3 Experimenting Channel Selection Functionality Using the IRIS Testbed

The experiment considered here focuses on learning the interference characterisation and using the learnt information for supporting channel selection in CRN. Specifically, the approach consists in analyzing the environment where a given cell (or access point) is operating by performing both radio-frequency and performance measurements and, based on these measurements, to characterise the observed interference in terms of the number of interfering sources. To support this knowledge discovery, the capabilities of the IRIS testbed are extended through the inclusion of the RapidMiner tool [10]. It is a powerful all-in-one tool that features hundreds of pre-defined data preparation and machine learning algorithms to support data science projects.

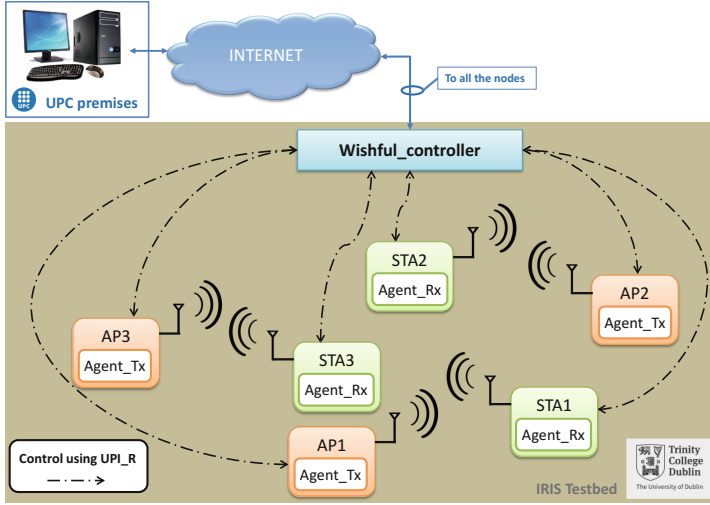


Fig. 1. Example of experimentation scenario

### 3.1 Learning Interference Characterisation

The example in Fig. 1 illustrates a scenario for learning interference characterization. Let us assume, as an example, that the receiver STA1 is connected to the transmitter AP1 operating at a given frequency. Simultaneously, the other transmitters (i.e. AP2 and AP3) may be operating in the same frequency, thus generating interference to STA1, or they may be operating in a different frequency, thus not generating interference. In this scenario, the objective of the considered experiment is to apply machine-learning based tools to smartly process the measurements performed by STA1 in order to characterize the existing interference. More specifically, it is proposed to use a supervised classification mechanism to estimate, based on the measurements of STA1, the number of interfering sources at a certain instant of time.

The classification is the process of finding a model or function that describes and distinguishes data classes or concepts. The obtained model (i.e. the classifier) is then used to determine the class to which an object belongs. The object is the entity to be classified and it is usually represented by a tuple that includes a set of attribute values (e.g. a tuple could be a set of measurements performed by a receiver and each of the measurements is an attribute). The classification process assumes that the possible classes are predefined in advance. Then, the classifier model is usually obtained from a supervised learning algorithm that analyses a set of training tuples associated with known classes.

Figure 2 illustrates the classification process. In general terms, the classifier takes as input a tuple of the form  $\mathbf{X}_t = \{x_{t,1}, x_{t,2}, \dots, x_{t,M}\}$  with  $M$  different measurements performed by a receiver at time  $t$ . The objective of the classifier is to make an association between the input tuple  $\mathbf{X}_t$  and the class  $C(\mathbf{X}_t)$  that specifies the number of interfering sources at time  $t$ . For that purpose the process involves the following steps:

1. Training stage (off-line operation): The classification model is initially obtained by means of a training stage consisting of a supervised learning process. The training stage uses as input  $S$  different tuples  $X_j$   $j = 1, \dots, S$  composed of measurements performed under interference conditions that are known *a priori*, meaning that the number of interferers, i.e. the class of each training tuple  $C(X_j)$ , is known during the measurements. These tuples and their associated classes are used as inputs to the training algorithm that will build the internal structure of the classifier. The specific training algorithm depends on the considered classification tool. The following alternatives are considered in this study [12]: decision tree, naive Bayes classifier, SVM and neural network.
2. Classification stage (on-line operation): The classification model obtained in the training stage is used to estimate the number of interferers for any tuple  $X_t = \{x_{t,1}, x_{t,2}, \dots, x_{t,M}\}$  with the measurements obtained at a certain time  $t$ .

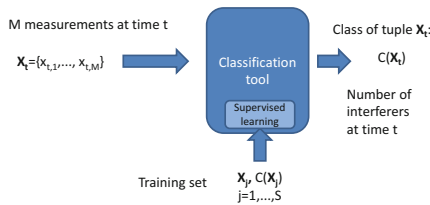


Fig. 2. Classification process

In the specific experiment on the IRIS testbed, we initially create tuples  $X_t$  with measurements of the throughput ( $Th$ ) and Received Signal Strength Indicator ( $RSSI$ ) at time  $t$ , i.e.  $X_t = \{Th(t), RSSI(t)\}$  under different interference situations (with 0 interferers, 1 interferer and 2 interferers). During the training stage, each of these tuples and the number of interferers for each one are used to build a classification model. Then, during the classification stage, the model is used each time that the methodology needs to estimate the number of interferers for each new tuple of measurements.

### 3.2 Channel Selection

Channel selection (also denoted as carrier selection) is the mechanism used to decide the operating channel (i.e. center frequency and associated bandwidth) of a transmitter. A smart channel selection mechanism is relevant to facilitate the coexistence between multiple transmitters in wireless scenarios operating in unlicensed spectrum when there is little or no coordination between these transmitters. This could be the case of e.g. Wi-Fi networks or unlicensed LTE (LTE-U).

The design of a proper channel selection functionality can greatly improve the overall efficiency of a wireless system when using unlicensed spectrum, since it will impact on the overall interference experienced by the receivers and thus on the achieved throughput performance.

Under the above considerations, the purpose of the experiment considered here is to use the IRIS testbed to assess a channel selection algorithm (Algorithm 1) that exploits

the extracted knowledge from the supervised classification process for characterizing the interference as explained in Sect. 3.1. For benchmarking purposes, a channel selection algorithm using Q-learning (Algorithm 2) and another one using game theory (Algorithm 3) have also been tested.

The general scenario assumes a total of  $T$  transmitters with their associated receivers and a total of  $K$  possible frequency channels. The considered channel selection algorithms are described in the following:

#### **Algorithm 1: Supervised Classification-Based Channel Selection Algorithm**

For the supervised classification-based channel selection algorithm for the  $i$ -th transmitter,  $i = 1, \dots, T$ , it is assumed that the training stage explained in Sect. 3.1 has been executed previously to build the classifier. Then, each time step, the receiver measures the values of throughput and  $RSSI$  for all the channels. Then, the classifier estimates the number of interferers in each of the channels. The estimated number of interferers is averaged considering a time window of  $N$  samples. The selected channel will be the one with minimum number of interferers. The process is subsequently repeated at the next time steps to consider possible changes in the environment (e.g. due to channel selections made by other transmitters) which could lead to new channel changes.

#### **Algorithm 2: Q-Learning-Based Channel Selection Algorithm**

Q-learning is a type of Reinforcement Learning (RL) technique [13] where learning is achieved through the interaction with the environment, so that the learner discovers which actions yield the most reward by trying them. In this way, each transmitter progressively learns and selects the channels that provide the best performance based on the previous experience. In the considered algorithm, described in detail in [14, 15], each transmitter  $i$  stores a value function  $Q(i, k)$  that measures the expected reward (i.e. throughput) that can be achieved by using each channel  $k$  according to the past experience. Whenever a channel  $k$  has been used by the transmitter  $i$ ,  $Q(i, k)$  is updated following a single state Q-learning approach with null discount rate and learning rate  $\alpha_L$ . Based on this, the channel selection decision-making follows the softmax policy with temperature  $\tau$ .

#### **Algorithm 3: Game Theory-Based Channel Selection Algorithm**

In this algorithm, the channel selection problem is modelled as a game in which each transmitter/receiver pair is a player and the actions made by each player are the selected channels. Specifically, here we consider the Iterative Trial and Error Learning-Best Action (ITEL-BA) algorithm described in [16]. In ITEL-BA, each transmitter retains a benchmark action  $a_{B,i}(t)$  (i.e. a benchmark channel to select) and the corresponding benchmark reward  $r_{B,i}(t)$  as a reference to evolve the action selection strategy. The reward is measured as the obtained throughput averaged during a time window of  $N$  samples. At a certain time, a channel is chosen depending on the so-called mood of the player, which basically captures the degree of satisfaction of the player with the current benchmark action and benchmark reward. The mood  $m_i(t)$  of player  $i$  at the beginning of time step  $t$  can be *content*, *discontent*, *hopeful* or *watchful*. The general idea is that a content player will be selecting the benchmark action most of the time, and will occasionally experiment with new actions according to a probability  $\epsilon \ll 1$  called exploration rate. Instead, a discontent player will try out new actions frequently, eventually

becoming content. The *hopeful* and *watchful* moods correspond to transitional situations, triggered by changes in the behavior of other players (or in the environment), and they will facilitate updates in the values of the benchmark action and reward to cope with these changes. The reader is referred to [16] for a detailed specification of the ITEL-BA algorithm.

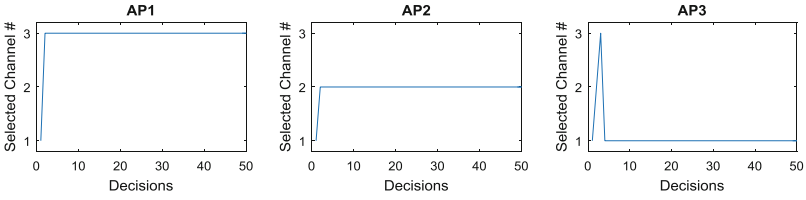
## 4 Results

The evaluation of the channel selection algorithms is performed using the set-up of the IRIS testbed illustrated in Fig. 1. It is considered that 3 nodes act as APs (AP1, AP2, AP3). Each APs has an associated receiver (STA1, STA2, STA3). There are 3 possible channels to select: Channel #1: 2890 MHz, Channel #2: 2900 MHz and Channel #3: 2910 MHz.

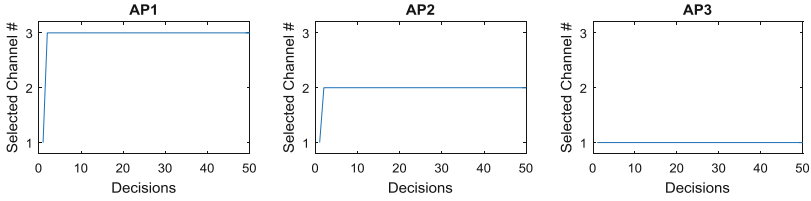
Initially, all the APs transmit on Channel #1. Subsequently, each AP can change channel being used according to the different channel selection algorithms explained in Sect. 3.2.

### 4.1 Algorithm 1: Supervised Classification-Based Channel Selection

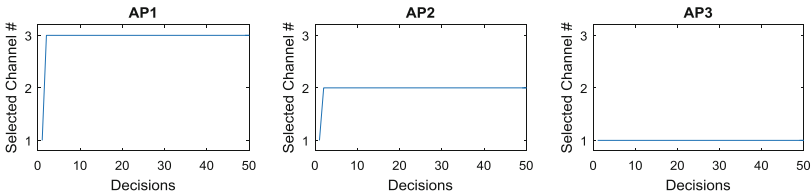
Different executions are performed for each of the considered classifiers. The algorithm is tested with an averaging window of  $N = 50$  samples. The results shown in Figs. 3, 4, 5 and 6 depict the channel number selected by each AP as a function of the number of channel selection decisions for the decision tree, naive Bayes, SVM and neural network classifiers, respectively. It is observed that, although all the APs start with the same Channel #1, in all the cases the APs are able to switch to a channel that is estimated by the classifier to be free of interferers. As a result, the system is able to find an optimum configuration in which each AP uses a different channel and correspondingly there is no interference. It is also worth observing that the naive Bayes and SVM classifiers are able to switch to a channel free of interferers very quickly, in just one channel selection decision. In the decision tree, naive Bayes and SVM classifiers, AP1 switches to Channel #3, AP2 switches to Channel #2 and AP3 remains in the same Channel #1. This solution is kept for the rest of the execution and no further changes are performed. In turn, focusing on the behavior of the decision tree classifier (see Fig. 3), it is observed that, due to the lower accuracy of this classifier, it requires a few more decisions to reach the optimum configuration in which each AP uses a different frequency. For example, it is observed that, at the beginning, AP3 makes a wrong decision by switching temporarily to Channel #3, which is being used by AP1, but then it moves to Channel #1. As for the neural network classifier, which also has lower accuracy, Fig. 6 reflects that, at the beginning, the APs quickly find a solution with different channels (i.e. AP1 using Channel #3, AP2 using Channel #2 and AP3 using Channel #1). However, after some time, AP2 makes a wrong decision and switches to the Channel #1 used by AP3. This situation is solved after 10 further decisions, when AP3 switches to Channel #2.



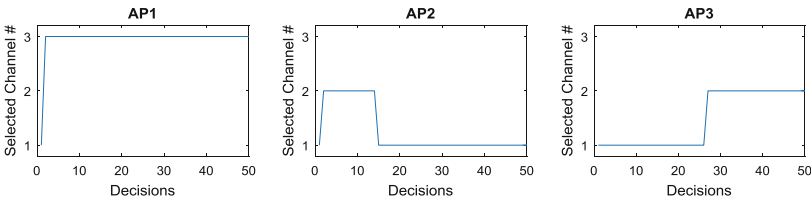
**Fig. 3.** Selected channel with Algorithm 1 and Decision Tree classifier for each AP



**Fig. 4.** Selected channel with Algorithm 1 and Naive Bayes classifier for each AP



**Fig. 5.** Selected channel with Algorithm 1 and SVM classifier for each AP



**Fig. 6.** Selected channel with Algorithm 1 and Neural Network classifier for each AP

## 4.2 Algorithm 2: Q-Learning-Based Channel Selection

The set-up for this execution is the same as for Algorithm 1, with all the three APs working initially in Channel #1. The Q-learning algorithm is configured with learning rate  $\alpha_L = 0.1$ , while the temperature parameter  $\tau$  is initially 0.15 and is reduced in each decision following a logarithmic cooling approach as explained in [14]. Figure 7 depicts the evolution of the channels selected by each AP with the successive channel selection decisions. It is observed that after some fluctuations associated to the probabilistic behavior of the softmax decision-making criterion finally the experiment converges to a solution where each AP has selected a different channel. Specifically, after



convergence AP1 operates with Channel #2, AP2 with Channel #1 and AP3 with Channel #3. The maximum number of decisions taken by an AP before converging in this case is 15.

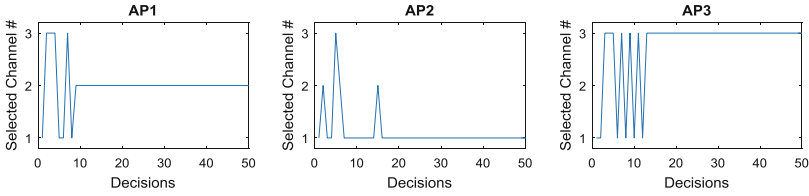


Fig. 7. Selected channel numbers with Algorithm 2 (Q-learning) for each AP

### 4.3 Algorithm 3: Game Theory-Based Channel Selection

Again, the set-up of the network is the same as in the previous cases. The game theory-based algorithm is configured with an averaging window of  $N = 50$  samples and exploration rate  $\varepsilon = 0.01$ . Figure 8 represents the evolution of the channel selected by each AP as a function of the number of channel selection decisions. It can be observed how this algorithm is also able to converge to an optimum solution where all the APs operate in a different channel, i.e. AP1 in Channel #3, AP2 in Channel #1 and AP3 in Channel #2. In this case, the maximum number of decisions made by an AP before reaching the optimum solution is 27 (for the case of AP1).

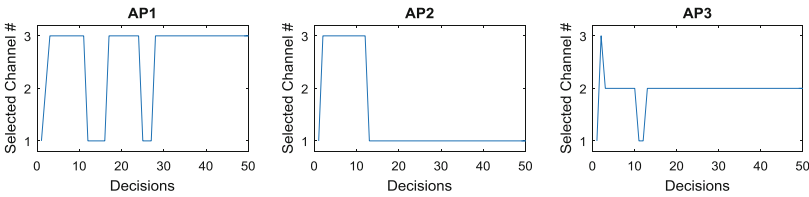


Fig. 8. Selected channel numbers with Algorithm 3 (game theory) for each AP

## 5 Conclusions

This paper has presented an experiment focusing on the channel selection functionality for Cognitive Radio Networks (CRN), so that an access point decides the most appropriate channel to use within a band that is shared among multiple transmitters. This selection has been based on a supervised classification that allows estimating the number of interfering sources existing in a given frequency channel. Specifically four different classifiers have been considered: decision tree, neural network, naive Bayes and Support Vector Machine (SVM). The channel selection algorithm exploits the estimation of the number of interferers to decide the most convenient channel to be used by a transmitter. Furthermore, a comparison against other Channel Selection strategies using Q-learning and game theory-based mechanisms has also been performed. Results in a scenario with

3 pairs of transmitter/receiver APs have revealed that all the considered algorithms for channel selection converge to an optimum solution where all the pairs operate in a different channel. Furthermore, it has been observed that the fastest convergence is achieved with the SVM and Naive Bayes classifiers, while the Game Theory and Q-learning based approaches exhibit slower convergence.

**Acknowledgements.** This work has been done using the IRIS testbed available through the WiSHFUL Platform. This work has been partly supported by the Spanish Research Council and FEDER funds under RAMSES and SONAR 5G grants (ref. TEC2013-41698-R and TEC2017-82651-R).

## References

1. Mitola III, J.: Cognitive radio: an integrated agent architecture for software defined radio. Ph.D. dissertation, KTH Royal Institute of Technology (2000)
2. El Hattachi, R., Erfanian, J. (eds.): NGMN 5G White Paper. NGMN Alliance, February 2015. [https://www.ngmn.org/fileadmin/ngmn/content/down-loads/Technical/2015/NGMN\\_5G\\_White\\_Paper\\_V1\\_0.pdf](https://www.ngmn.org/fileadmin/ngmn/content/down-loads/Technical/2015/NGMN_5G_White_Paper_V1_0.pdf)
3. Chih-Lin, I., Liu, Y., Han, S., Wang, S., Liu, G.: On big data analytics for greener and softer RAN. IEEE Access, August 2015. <http://ieeexplore.ieee.org/document/7210136/>
4. Pérez-Romero, J., Sánchez-González, J., Sallent, O., Agustí, R.: On learning and exploiting time domain traffic patterns in cellular radio access networks. Machine Learning and Data Mining in Pattern Recognition. LNCS (LNAI), vol. 9729, pp. 501–515. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-41920-6\\_40](https://doi.org/10.1007/978-3-319-41920-6_40)
5. Sánchez-González, J., Pérez-Romero, J., Agustí, R., Sallent, O.: On learning mobility patterns in cellular networks. In: Iliadis, L., Maglogiannis, I. (eds.) AIAI 2016. IAICT, vol. 475, pp. 686–696. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-44944-9\\_61](https://doi.org/10.1007/978-3-319-44944-9_61)
6. Sánchez-González, J., Sallent, O., Pérez-Romero, J., Agustí, R.: On extracting user-centric knowledge for personalised quality of service in 5G networks. In: IFIP/IEEE International Symposium on Integrated Network Management - 2nd International Workshop on Analytics for Network and Service Management, Lisbon, Portugal, May 2017
7. WiSHFUL. <http://www.wishful-project.eu/>
8. IRIS Testbed. <https://iris-testbed.connectcentre.ie>
9. D10.1 Design of software architecture for intelligent control and showcases. [http://www.wishful-project.eu/sites/default/files/images/WiSHFUL\\_D10.1\\_Lead\\_iMinds\\_R\\_PU\\_2015-12-23\\_Final.pdf](http://www.wishful-project.eu/sites/default/files/images/WiSHFUL_D10.1_Lead_iMinds_R_PU_2015-12-23_Final.pdf)
10. RapidMiner. <https://rapidminer.com/>
11. Pérez-Romero, J., Sallent, O., Ferrús, R., Agustí, R.: Knowledge-based 5G radio access network planning and optimization. In: International Symposium on Wireless Communication Systems (ISWCS), Poznan, Poland, August 2016
12. Han, J., Kamber, M.: Data Mining Concepts and Techniques, 2nd edn. Elsevier, New York City (2006)
13. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
14. Sallent, O., Pérez-Romero, J., Ferrús, R., Agustí, R.: Learning-based coexistence for LTE operation in unlicensed bands. In: International Conference on Communications (Workshops), IEEE ICC 2015, 8–12 June 2015, London, United Kingdom. pp. 2307–2313 (2015)

15. Perez-Romero, J., Sallent, O., Ferrús, R., Agusti, R.: A robustness analysis of learning-based coexistence mechanisms for LTE-U operation in non-stationary conditions. In: 82nd Vehicular Technology Conference, IEEE VTC 2015-Fall, 6–9 September 2015, Boston, MA, pp. 1–5 (2015)
16. Pérez-Romero, J., Sallent, O., Ahmadi, H., Macaluso, I.: On modeling channel selection in LTE-U as a repeated game. In: Wireless Communications and Networking Conference, IEEE WCNC 2016, 3–6 April 2016, Doha, Qatar (2016)