

A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks

Paulo Valente Klaine, *Student Member, IEEE*, Muhammad Ali Imran, *Senior Member, IEEE*, Oluwakayode Onireti, *Member, IEEE*, and Richard Demo Souza, *Senior Member, IEEE*

Abstract—In this paper, a survey of the literature of the past 15 years involving machine learning (ML) algorithms applied to self-organizing cellular networks is performed. In order for future networks to overcome the current limitations and address the issues of current cellular systems, it is clear that more intelligence needs to be deployed so that a fully autonomous and flexible network can be enabled. This paper focuses on the learning perspective of self-organizing networks (SON) solutions and provides, not only an overview of the most common ML techniques encountered in cellular networks but also manages to classify each paper in terms of its learning solution, while also giving some examples. The authors also classify each paper in terms of its self-organizing use-case and discuss how each proposed solution performed. In addition, a comparison between the most commonly found ML algorithms in terms of certain SON metrics is performed and general guidelines on when to choose each ML algorithm for each SON function are proposed. Lastly, this paper also provides future research directions and new paradigms that the use of more robust and intelligent algorithms, together with data gathered by operators, can bring to the cellular networks domain and fully enable the concept of SON in the near future.

Index Terms—Machine learning, self-organizing networks, cellular networks, 5G.

I. INTRODUCTION

BY 2020, it is expected that mobile traffic will grow around ten thousand times of what it is today and that the number of devices connected to the network will be around fifty billion [1]–[3]. Because of the exponential growth that is expected in both connectivity and density of traffic, primarily due to the advances in the Internet of Things (IoT) domain, Machine-to-Machine (M2M) communications, cloud computing and many other technologies, 5G will need to push the network performance to a next level. Furthermore, 5G will also have to address current limitations of Long Term Evolution (LTE) and LTE-Advanced (LTE-A), such as

latency, capacity and reliability. Some of the requirements that are recurrent in state-of-the-art literature for 5G networks are [4]–[6]:

- Address the growth required in coverage and capacity;
- Address the growth in traffic;
- Provide better Quality of Service (QoS) and Quality of Experience (QoE);
- Support the coexistence of different Radio Access Network (RAN) technologies;
- Support a wide range of applications;
- Provide peak data rates higher than 10 Gbps and a cell-edge data rate higher than 100 Mbps;
- Support radio latency lower than one millisecond;
- Support ultra high reliability;
- Provide improved security and privacy;
- Provide more flexibility and intelligence in the network;
- Reduction of CAPEX and OPERational EXpenditures (CAPEX and OPEX);
- Provide higher network energy efficiency.

As it can be seen, all of these requirements are very stringent. Hence, in order to meet these requirements, new technologies will have to be deployed in all network layers of the 5G network. Several breakthroughs are being discussed in the literature in the past couple of years, the most common ones being: massive MIMO (Multiple-Input Multiple-Output), millimeter-waves (mmWaves), new physical layer waveforms, network virtualization, control and data plane separation, network densification (deployment of several small cells) and implementation of self-organizing Networks (SON) functions [5].

Although all of these breakthroughs are very important and often referred as a necessity for future mobile networks, the concept of network densification is the one that will require heavier changes in the network and possibly a change in paradigm in terms of how network solutions are provided [7]. In addition, the deployment of several small cells would most likely address the current limitations of coverage, capacity and traffic demand, while also providing higher data rates and lower latency to end users [5].

While the densification will result in all these benefits, it will also generate several new problems to the operators in terms of coordination, configuration and management of the network. The dense deployment of several small cells, will result in an increase in the number of mobile nodes that will need to be managed by mobile operators. Furthermore, these types of cells will also collect an immense

Manuscript received January 30, 2017; revised May 23, 2017; accepted July 11, 2017. Date of publication July 17, 2017; date of current version November 21, 2017. This work was supported in part by the DARE Project through the Engineering and Physical Sciences Research Council U.K. Global Challenges Research Fund Allocation under Grant EP/P028764/1, and in part by the Conselho Nacional de Pesquisa, Brazil, under Grant 304096/2013-0. (Corresponding author: Paulo Valente Klaine.)

P. V. Klaine, M. A. Imran, and O. Onireti are with the School of Engineering, University of Glasgow, Glasgow G12 8QQ, U.K. (e-mail: p.valente-klaine.1@research.gla.ac.uk; muhammad.imran@glasgow.ac.uk; oluwakayode.onireti@glasgow.ac.uk).

R. D. Souza is with the Federal University of Santa Catarina, Florianópolis 88040900, Brazil (e-mail: richard.demo@ufsc.br).

Digital Object Identifier 10.1109/COMST.2017.2727878

amount of data in order to monitor network performance, maintain network stability and provide better services. This will result in an increasingly complex task just to configure and maintain the network in an operable state if current techniques of network deployment, operation and management are applied [8].

One possible way of solving these issues is by deploying more intelligence in the network. The main objectives of SON are to provide intelligence inside the network in order to facilitate the work of operators, provide network resilience, while also reducing the overall complexity, CAPEX and OPEX, and to simplify the coordination, optimization and configuration procedures of the network [9], [10].

A. Overview of Self-Organizing Networks

SON can be defined as an adaptive and autonomous network that is also scalable, stable and agile enough to maintain its desired objectives [10]. Hence, these networks are not only able to independently decide when and how certain actions will be triggered, based on their continuous interaction with the environment, but are also able to learn and improve their performance based on previous actions taken by the system. The concept of SON in mobile networks can also be divided into three main categories. These categories are: self-configuration, self-optimization and self-healing and are commonly denoted jointly as *self-x* functions [10].

Self-configuration can be defined as all the configuration procedures necessary in order to make the network operable. These configuration parameters can come in the form of individual Base Station (BS) configuration parameters, such as IP configuration, Neighbor Cell List (NCL) configuration, radio and cell parameters configuration or configurations that will be applied to the whole network, such as policies. Self-configuration is mainly activated whenever a new base station is deployed in the system, but it can also be activated if there is a change in the system (for example, a BS failure or change of service or network policies).

After the system has been correctly configured, the self-optimization function is triggered. The self-optimization phase can be defined as the functions which continuously optimize the BSs and network parameters in order to guarantee a near optimal performance. Self-optimization can occur in terms of backhaul optimization, caching, coverage and capacity optimization, antenna parameters optimization, interference management, mobility optimization, HandOver (HO) parameters optimization, load balancing, resource optimization, Call Admission Control (CAC), energy efficiency optimization and coordination of SON functions. By monitoring the system continuously, and using reported measurements to gather information, self-optimization functions can ensure that the objectives are maintained and that the overall performance of the network is near optimum.

In parallel to self-optimization, the function of self-healing can also be triggered. Since no system is perfect, faults and failures can occur unexpectedly and it is no different with cellular systems. Whenever a fault or failure occurs, for whatever reason (e.g., software or hardware malfunction) the

self-healing function is activated. Its objective is to continuously monitor the system in order to ensure a fast and seamless recovery. Self-healing functions should be able not only to detect the failure events but also to diagnose the failure (i.e., determine why it happened) and also trigger the appropriate compensation mechanisms, so that the network can return to function properly. Self-healing in cellular systems can occur in terms of network troubleshooting (fault detection), fault classification, and cell outage management [10]–[13].

Also, each SON function can be divided into sub-sections, commonly known as use-cases. Figure 1 shows an outline of the most common use cases of each SON task. As it can be seen from Fig. 1, future cellular networks are expected to address several different use cases and provide many solutions in domains that either do not exist today or are beginning to emerge. Current methods today lack the adaptability and flexibility required to become feasible solutions to 5G networks. Although mobile operators collect a huge amount of data from the network in the form of network measurements, control and management interactions and even data from their subscribers, current methods applied to configure and optimize the network are very rudimentary. Such methods consist of manual configuration of thousands of BS parameters, periodic drive tests and analysis of measurement reports in order to operators to continuously optimize the network [9]. Furthermore, operators also require skilled personnel in order to constantly observe alarms and use monitoring software at the Operation and Management Center (OMC) to preform self-healing.

Many of the solutions require expert engineers to analyze data and adjust system parameters manually in order to optimize or configure the network. Some other solutions also require expert personnel on site in order to fix certain problems, when detected. All these solutions are extremely ineffective and costly to mobile operators and, although operators collect a huge amount of mobile data daily, it is not being used at its full potential.

In order to leverage all the information that is already collected by operators and provide the network with adaptable and flexible solutions, it is clear that more intelligence needs to be deployed. With that in mind, several Machine Learning (ML) solutions are being applied in the context of SON to explore the different kinds of data collected by operators. Thus, a SON system, in order to be able to perform all three functions, needs some sort of intelligence. This paper provides an extensive literature review of the ML algorithms that are being applied in mobile cellular SON, in order to achieve its objectives in each of the *self-x* functions.

B. Machine Learning in SON

Despite being in its infancy, the concept of SON in mobile cellular networks is developing really fast. Several research groups are implementing intelligent solutions to address certain use cases of mobile networks and also standardize some methods, as it can be seen from The 3rd Generation Partnership Project (3GPP), Next Generation Mobile Networks (NGMN) Alliance, mobile operators and many other research initiatives. Current state-of-the-art

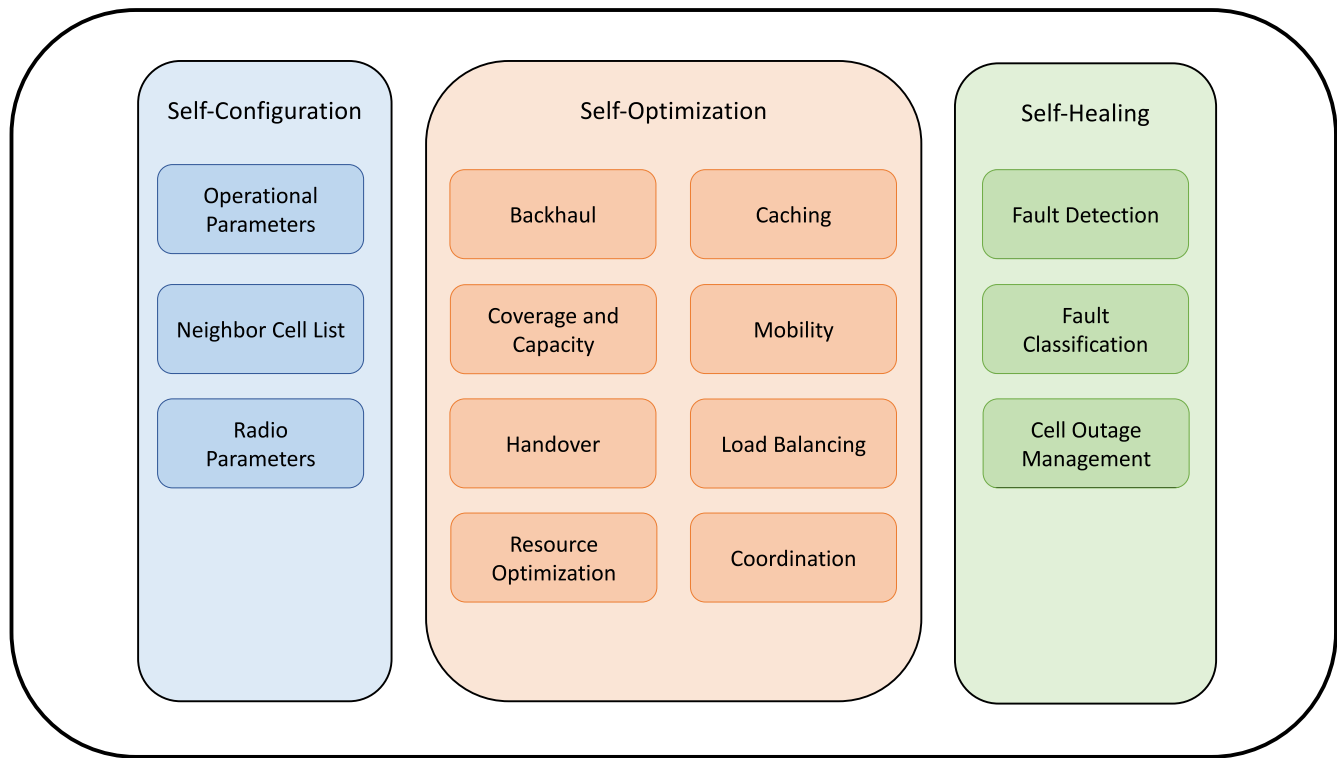


Fig. 1. Major use cases of each SON function: self-configuration, self-optimization and self-healing.

algorithms go all the way from basic control loops and threshold comparisons to more complex ML and data mining techniques [14]. As the field develops, there is a significant trend of implementing more robust and advanced techniques which would in turn solve more complex problems [15]. This paper also provides a basic overview of the current state-of-the-art ML techniques that are being developed and applied to cellular networks.

The main categories that ML algorithms can be fitted into are supervised, unsupervised and Reinforcement Learning (RL). Supervised learning, as the name implies, requires a supervisor in order to train the system. This supervisor tells the system, for each input, what is the expected output and the system then learns from this guidance. Unsupervised learning, on the other hand, does not have the luxury of having a supervisor. This occurs, mainly when the expected output is not known and the system will then have to learn by itself. Lastly, RL works similarly to the unsupervised scenario, where a system must learn the expected output on its own, but on top of that, a reward mechanism is applied. If the decision made by the system was good, a reward is given, otherwise the system receives a penalty. This reward mechanism enables the RL system to continuously update itself, while the previous two techniques provide, in general, a static solution.

However, as it will be seen in the upcoming sessions of the paper, several other techniques like Markov models, heuristics, fuzzy controllers and genetic algorithms are also being applied to provide intelligence to cellular networks. One problem that arises, however, is that as the techniques get more complex, more data is required for the algorithm to perform

well. That is why the concept of Big Data is also inter-linked with SON, so that the ML algorithms can work to their full potential [8], [9], [16]–[18]. With the deployment of SON together with big data, the huge amount of data gathered by mobile operators will become more useful and new applications and innovative solutions, such as participatory sensing, can be enabled [19].

C. Paper Objectives and Contributions

As previously stated, one of the objectives of this survey is to provide an extensive literature review over the past fifteen years on efforts to implement intelligent solutions in the realm of cellular networks, in order to automate and manage an increasingly more complex and developing network. The paper covers not only the recent research that is related to SON, but also previous research carried out that involved ML algorithms and implementations of automated functions that improved the overall performance of cellular networks.

In contrast to other surveys in the area, such as [10], which focused on introducing readers to the concept of SON in cellular networks, its definitions, applications and use cases, [20], which focused on basic definitions and concepts of self-organizing systems and how could self-organization be applied in the context of wireless sensor networks, or even [21], which focused on different types of self-organizing networks applied in the domains of wireless sensor networks, mesh networks and delay tolerant networks, this paper surveys the application of ML algorithms in cellular networks, and, much

like [22] and [23], it provides a more in-depth view of how and why each intelligent technique is applied.

However, differently from [22] and [23], which surveyed the application of ML algorithms in cognitive radios and wireless sensor networks, respectively, this survey is applied in the domain of cellular networks and discusses how each technique can be applied in terms of each SON function. Based on that, it is assumed that the reader already is familiar with SON concepts and its use cases, otherwise the reader can refer to [10].

In addition to that, this work also provides a short tutorial and explanation of the most popular ML solutions that are being applied in the realm of cellular networks, so that readers that are interested in applying these algorithms can have a basic knowledge of how they work and when they should be used. Last but not least, another objective of the paper is to explore new research directions and propose new solutions to current SON problems, in order to achieve more automation and intelligence in the network.

The main contributions of this paper are:

- To provide the readers with an extensive overview of the literature involving SON applied to cellular networks and the most popular ML algorithms and techniques involved when implementing SON functions;
- The paper focus is on the learning perspective of ML algorithms applied to SON. Instead of providing an overview of SON functions, this paper contribution is more related to provide the readers with an understanding and classification of the state-of-the-art algorithms implemented to achieve these SON functions;
- The paper also tries to categorize each algorithm according to their SON function and ML implementation;
- The paper also proposes to classify different algorithms based on their learning and technique applied, mainly: supervised, unsupervised, controllers, RL, Markov models, heuristics, dimension reduction and Transfer Learning (TL);
- Compare different ML techniques in terms of some SON requirements;
- Provide general guidelines on when to use each ML algorithm for each SON function;

A complete list of acronyms can be found in Table I and the remainder of this paper is structured as follows: Section II provides a brief tutorial of the most popular learning techniques used to address SON use cases. Sections III–V define the learning problem in self-configuration, self-optimization and self-healing, respectively and each section explains how learning can be applied within each category. Section VI analyses the most common ML technique applied in cellular SON and discusses their strengths, weaknesses and also discusses which ML algorithm is more suitable for each SON function. Section VII provides future research directions and suggestions of new implementations and Section VIII concludes the paper.

TABLE I
LIST OF ACRONYMS

Symbol	Description
AD	Anomaly Detector
AIW-PSO	Adaptive Inertia Weight Particle Swarm Optimization
BMU	Best Matching Unit
BS	Base Station
CAC	Call Admission Control
CAPEX	CAPital EXPenditures
CAT	Cell-Aware Transfer
CDMA	Code Division Multiple Access
CF	Collaborative Filtering
CRE	Cell Range Extension
CSI	Channel State Information
D2D	Device-to-Device
DM	Diffusion Maps
DNS	Domain Name System
DT	Decision Trees
ELF	Evolutionary Learning of Fuzzy
EMD	Entropy Minimization Discretization
ESM	Energy Saving Mechanism
FLC	Fuzzy Logic Controller
FQL	Fuzzy Q-Learning
GA	Genetic Algorithm
HMM	Hidden Markov Model
HO	HandOver
HOAP	HandOver Aggregate Performance
ICIC	Inter-Cell Interference Coordination
IoT	Internet of Things
k -NN	K -Nearest Neighbors
KPI	Key Performance Indicator
LOFAD	Local Outlier Factor based Anomaly Detector
LTE	Long Term Evolution
LTE-A	Long Term Evolution - Advanced
M2M	Machine-to-Machine
MC	Markov Chains
MCA	Minor Component Analysis
MDS	MultiDimensional Scaling
MDT	Minimization of Drive Test
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MLB	Mobility Load Balancing
MRO	Mobility Robustness Optimization
NCL	Neighbor Cell List
NN	Neural Network
OAM	Operation, Administration and Maintenance
OCSVMD	One Class Support Vector Machine based Detector
OFDMA	Orthogonal Frequency Division Multiple Access
OMC	Operation and Management Center
OPEX	OPERational EXPenditures
PCA	Principal Component Analysis
PCI	Physical Cell Identity
QL	Q-Learning
QoE	Quality of Experience
QoS	Quality of Service
RACH	Random Access CHannel
RAN	Radio Access Network
RL	Reinforcement Learning
RLF	Radio Link Failure
RSRP	Reference Signal Received Power
SINR	Signal to Interference plus Noise Ratio
SNR	Signal to Noise Ratio
SOM	Self Organizing Map
SON	Self Organizing Networks
SVM	Support Vector Machine
TL	Transfer Learning
TTT	Time-To-Trigger
VHO	Vertical HandOver

II. OVERVIEW OF MACHINE LEARNING ALGORITHMS

The concept of SON in cellular networks was defined in [10] as a network that not only has adaptive and autonomous

functions, but also is scalable, stable and agile enough in order to maintain its desired objectives even when changes occur in the environment. Although learning is not implicitly included

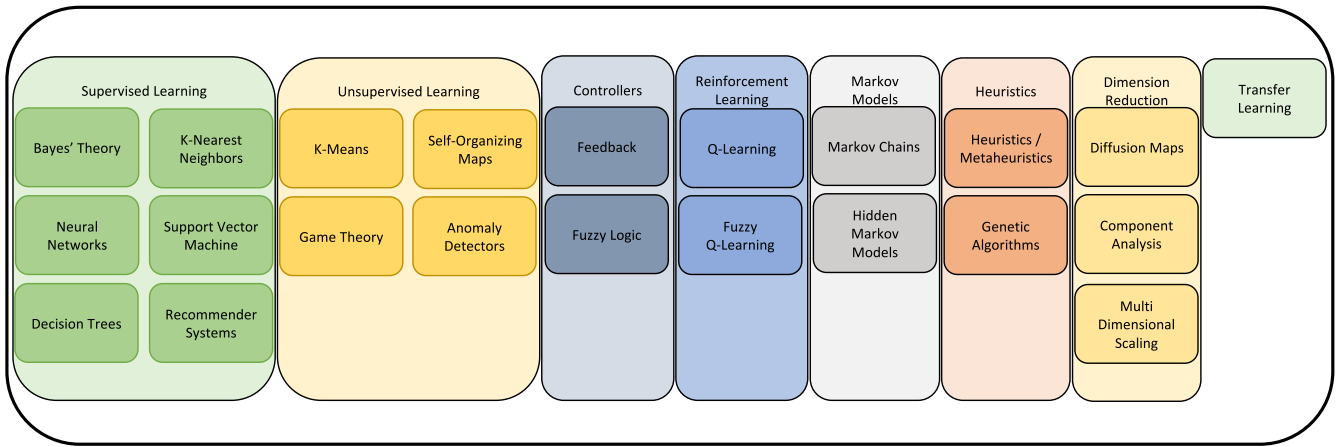


Fig. 2. Block diagram showing the most common algorithms in the literature of cellular SON and how they are classified.

in the SON definition, intelligence is crucial to a SON system in order to accomplish its objectives.

This section consists of basic tutorials on some of the most researched and applied intelligent algorithms to cellular network use cases. Each algorithm is briefly explained with some examples and some basic references are also provided for readers interested in further information about each technique.

However, before starting, let us begin by defining the main goals of ML and the basic categories of learning that will be found in this paper. According to [24], ML is the science of making computers take decisions without being explicitly programmed to. This is done by programming a set of algorithms that analyze a given set of data and try to make predictions about it. Depending on how learning is performed, these algorithms are classified differently. Figure 2 shows different learning schemes and how they are related to each other.

A. Supervised Learning

Supervised learning, as the name implies, is a type of learning that requires a supervisor in order for the algorithms to learn their parameters. In this type of learning, the algorithms are given a set of data which contains both input and output information. Based on the input-output relationship, a model for the data can be determined, and, after that, a new set of input data is gathered and fed into the learned model so that the algorithm can make its predictions [25], [26].

In the context of cellular networks, supervised learning can be applied in several domains, such as: mobility prediction [27]–[30], resource allocation [31]–[33], load balancing [34], HO optimization [35], [36], fault classification [37], [38] and cell outage management [39]–[42].

Supervised learning is a very broad domain and has several learning algorithms, each with their own specifications and applications. In the following, the most common algorithms applied in the context of cellular networks are presented.

1) *Bayes' Theory*: The Bayes' theorem is an important rule in probability and statistical analysis to compute conditional probabilities, i.e., to understand how the probability of a hypothesis (h) is affected in the light of a new evidence (e).

The Bayes theorem is given by

$$P(h|e) = \frac{P(e|h)P(h)}{P(e)}, \quad (1)$$

where $P(h|e)$ is the probability of hypothesis h being true, given the new evidence e , also known as the *posterior probability*, $P(e|h)$ is the *likelihood* of evidence e on the hypothesis h , $P(h)$ is the probability before the new evidence is taken into account, known as *prior probability* and $P(e)$ is the probability of evidence e [43].

Bayes' theory provided a new understanding of probabilities and its applications, hence it is widely used in a lot of different areas. In the context of cellular networks, Akoush and Sameh [44], for example, used Bayes' theorem together with neural networks in order to enhance its learning procedure and try to predict a mobile user's position.

Another area where Bayes' theory can be applied is in the area of classification. Bayes' classifiers are simple probabilistic classifiers based on the application of the Bayes' theorem. Also, one assumption that is often made is that the inputs are independent from one another. This assumption leads to the creation of Naive Bayes' Classifiers. Recent research has applied the concept of Bayes' classifiers in fault detection [45], and fault classification [37], [38]. For interested readers, a more in-depth review of Bayesian classifiers, its advantages and disadvantages, and its two models, can be found in [46] and [47].

2) *k-Nearest Neighbor (k-NN)*: Another popular method of supervised learning is k -NN. This algorithm is applicable to problems where the underlying joint distribution of the observation and the result is not known. The algorithm does a very simple process: it tries to classify a new sample based on how many neighbors of a certain class that unclassified sample has [26]. For example, if a certain number of samples, in this case k , near the unclassified sample belongs to class A, then it is most probable that the new sample also belongs to class A. Fig. 3 shows a simple example of how this process is done.

Since the k -NN algorithm main metric is the distance between the unlabeled sample and its closest neighbors, several distance metrics can be applied. The most common ones

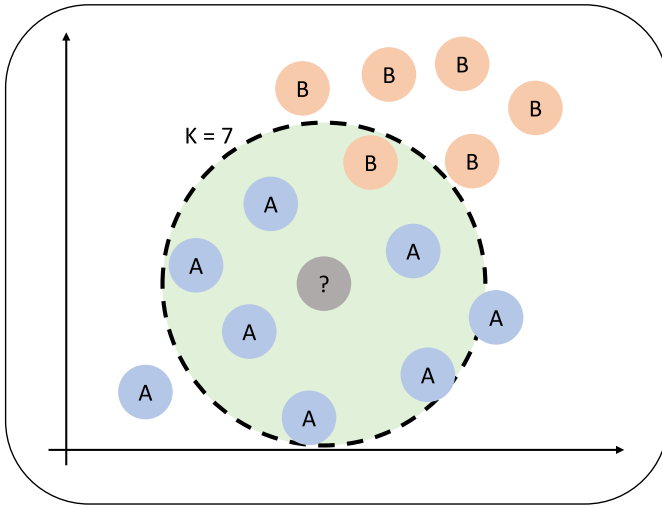


Fig. 3. Example of k -NN algorithm, for $k = 7$. In this case, the algorithm will decide that the unlabeled example should be classified as class A, since there are more neighbors from class A than class B closer to the unlabeled example.

are: Euclidean, Euclidean squared, City-block and Chebyshev. For more information on k -NN, please refer to [26] and [48].

K -NN can also be applied to solve regression problems, however, it is mostly used in the classification realm. In the case of cellular networks, k -NN is generally applied in the context of self-healing, either by detecting outage or sleeping cells [42], [49]–[52].

3) *Neural Networks (NNs)*: The concept of Neural Networks (NNs), also known as MultiLayer Perceptrons (MLPs), emerged as an attempt of simulating into computers the same behavior seen in the human brain. The human brain is a complex machine that performs highly complex, nonlinear and parallel computations all the time. However, by dividing these functions into very basic components, known as neurons, and by giving these neurons all the same computation function, a simple algorithm can become a very powerful computational tool.

The equivalent components of the neurons in a NN are its nodes. These nodes are responsible for performing nonlinear computations, by using their activation functions, and are connected to each other by variable link weights, which simulates the way neurons are connected in the human brain. These activation functions can vary depending on the design of the network, but the most frequent functions used are the sigmoid or the hyperbolic tangent functions [53].

The most basic design a NN can have is a network of three layers, consisting of an input layer, a hidden layer and an output layer. Although all networks must have an input and an output layer, the number of hidden layers or the number of nodes is not fixed. A simple NN design of three layers is shown on Fig. 4. As it can be seen from Fig. 4, the connections between different layers always go forward and do not form a cycle, therefore, this type of network is commonly known as feed forward neural network. There are other types of NNs, but this paper focuses only on feed forward NNs.

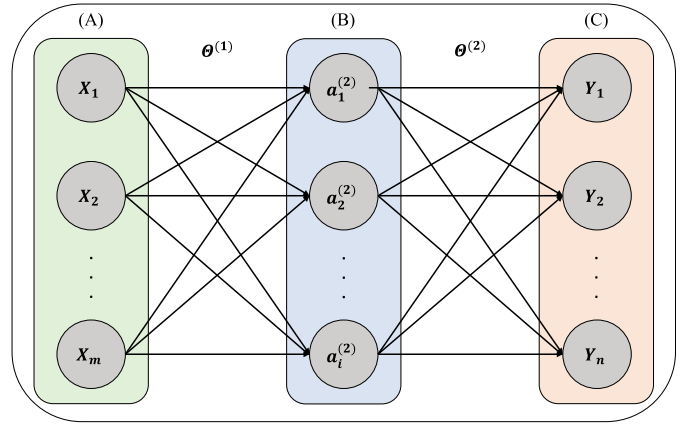


Fig. 4. Most basic design of a neural network, consisting of 3 layers, where (A) denotes the input layer, (B) the hidden layer and (C) the output layer. The inputs are denoted as x_1, \dots, x_m and outputs as y_1, \dots, y_n , where m denotes the total number of input features and n the total number of possible classes an input can be assigned to. Also, the variable link weights are depicted as $\Theta^{(j)}$, which correspond to the matrix of weights controlling the function mapping between layer j to layer $j+1$ and the activation function of each neuron as $a_i^{(j)}$, where i is the neuron number and j is the layer number.

By changing the number of nodes and the number of hidden layers, NNs can map highly complex functions and achieve very good performance. Hence, NN's are used in a wide range of applications. Another parameter that can be tuned in a NN is its learning method. Since the objective of the NN is to produce the best values of Θ (link weights) that maps the inputs to outputs, how the network learn this parameter can also be configured. The most common method used is the backpropagation method, but there are many others [53], such as Bayesian learning [26], [44], RL and random learning [33], [54]. Although NNs are not restricted to classification problems and can be used in nonlinear regression problems as well, most NNs are used as classifiers. For information about NNs in regression, please see [53].

In the context of cellular systems, NNs are applied specially in the self-optimization and self-healing scenarios, in terms of resource optimization [31]–[33], [55]–[58], mobility management [27], [28], [44], [59]–[63], HO optimization [35], [36], [64], [65], and cell outage management [41]. For more information about neural networks, how they work, basic properties and learning methods readers should go to [26], [53], and [66].

4) *Support Vector Machine (SVM)*: Another supervised learning technique commonly found in SON is the Support Vector Machine (SVM). The idea behind a SVM classifier is to map a set of inputs into a higher dimensional feature space. This is done through some linear or non-linear mapping and its objective is to maximize the distance between different classes. Since the goal of SVM is to find the hyperplane that produces the largest margin between different classes, SVM can also be known as a large margin classifier.

As the name implies, the SVM technique uses a subset of the training data as *support vectors* and they are crucial to the correct operation of this algorithm. In theoretical terms, the support vectors are the training samples that are closest to the decision surface and hence are the most difficult to

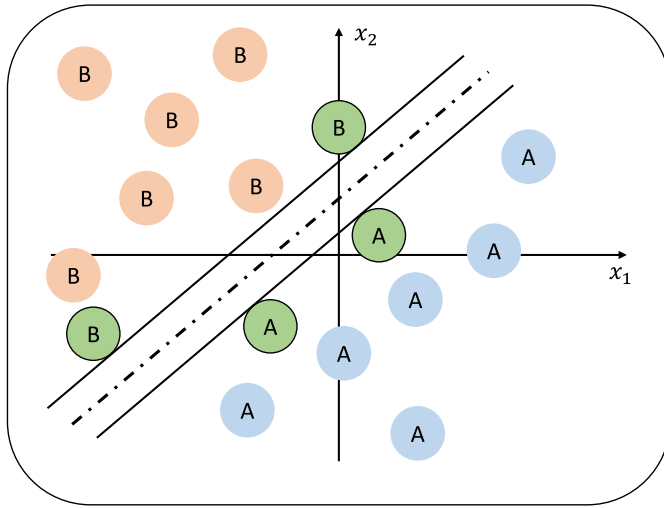


Fig. 5. An example of an SVM optimal linear hyperplane. The figure shows two classes, A and B, the green circles denote the support vectors and the shaded region denote the optimal decision boundary obtained. As it can be seen, by finding the largest margin between the two classes, the SVM algorithm determines the best decision region for each class.

classify. By finding the largest margin between these most difficult points, the algorithm can maximize the distance between classes and also guarantee that the decision region obtained for each class is the best one possible [43]. Figure 5 shows an example of an SVM classifier using linear mapping. For non-linear mapping, SVM can use different types of kernels, such as polynomial or Gaussian kernels. For a more thorough review of SVM, please refer to [26], [43], [53], and [67].

In the cellular networks domain, SVM is being applied in self-optimization and self-healing scenarios, more specifically in mobility optimization [30], [68], fault detection [69] and cell outage management [70], [71].

5) *Decision Trees*: Decision Trees (DT) are constructed by repeated splits of subsets of the original data into descendant subsets, however, despite being conceptually simple they are very powerful. The basic idea behind tree methods is that, based on the original data, a set of partitions is done so that the best class (in classification problems) or value (in regression problems) can be determined. The fundamental idea behind the partitions is to select each split so that the data contained on the descendant branches are “purer” than the data in the parent nodes [72].

In SON scenarios, tree algorithms are basically used to perform self-optimization and healing, either by performing mobility optimization [60], coordinating SON functions [73], detecting cell outage [40] or by doing classification of Radio Link Failures (RLFs) [74]. Figure 6 shows an example of a classification decision tree adapted from [74]. For more information on decision trees, please refer to [26] and [72].

6) *Recommender Systems*: Also known as *Collaborative Filtering* (CF) [75], [76], are a class of algorithms with the objective to provide suggestions for users based on the opinions of other users [77]. A simple example of recommendation algorithms are the suggestions made by e-commerce or video-based websites. The objective of a recommender system is to predict a set of items for the current user based

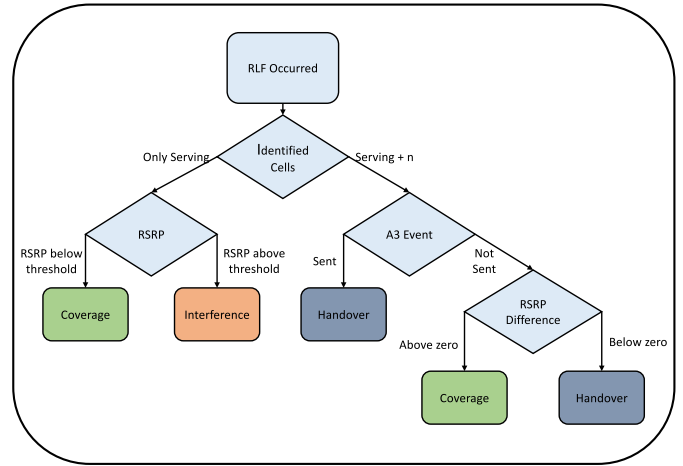


Fig. 6. An example of a decision tree classification adapted from [74]. In this problem, after a Radio Link Failure (RLF) occurred, the algorithm will try to identify the cause of the problem based on other measurements, such as Reference Signal Received Power (RSRP) and Signal to Interference plus Noise Ratio (SINR). Based on these measurements and comparing the RSRP with threshold and measuring its difference, the RLF events are then classified into one of three possible classes.

on a database of other users. There are two general classes of recommender algorithms: *memory-based* and *model-based* algorithms [78].

The *memory-based* algorithm tries to make predictions for a particular user based on the preferences of other users which are currently on the database’s memory. In addition, it also utilizes some knowledge about the current user, which can be of different items that the user has rated in the past. Together with this previous information, a set of weights is calculated from the user’s database and a prediction can be made. On the other hand, the *model-based* algorithm utilizes the user database as a reference and tries to build a model based on it. It then utilizes this model to predict a recommendation for the active user.

Recommender systems are very powerful and can be used in a wide range of applications. In cellular networks, most research is being done applying recommender systems to self-healing, more specifically to the cell outage management problem [79]–[81], but it can also be found on optimization of content caching [82].

B. Unsupervised Learning

In the case of unsupervised learning, an algorithm is given a set of inputs and its goal is to correctly infer the outputs without having a supervisor providing the correct answers or the degree of error for each observation. In other words, this learning method is given a set of unlabeled input data and it must correctly learn the outcomes [26]. Examples of unsupervised learning algorithms consist of clustering algorithms, combinatorial algorithms, Self-Organizing Maps (SOM), density estimation algorithms, Game Theory,¹ etc.

¹Game Theory can also be modeled as a RL problem, in which multiple agents take actions in a competing environment and at the end of the game, depending on each agent’s outcome, a reward or penalty is given.

Algorithm 1 *K-Means*

Input: Initial Data set (D), desired number of clusters (K)

- 1: Initialize K cluster centroids with random data points
- 2: **while** not converged **do**
- 3: For each center identify the closest data points
- 4: Compute means and assign new center
- 5: **end while**

In SON, unsupervised learning is applied in several domains, ranging from configuration of operational parameters [83], [84], caching [82], [85], [86], resource optimization [56], [87], [88], HO management [89], [90], mobility [91], load balancing [92], fault detection [93]–[102], cell outage management [49], [103]–[106], to sleeping cell management [50], [107].

Below is a review of the most popular unsupervised learning algorithms applied in SON.

1) *K-Means*: One of the most popular unsupervised learning algorithms found in the literature is *K-means*. This clustering algorithm is very useful in finding clusters and its centers in a set of unlabeled data. The algorithm is very simple and only requires two parameters: the initial data set and the desired number of clusters. The algorithm works as shown in Algorithm 1. As it can be seen, the algorithm is very easy and quick to deploy, hence its popularity. For more information on *K-means*, please refer to [26].

In SON, *K-means* can be found in mobility optimization [60], caching problems [82], resource optimization [56], [88], fault detection [108], and cell outage management [103].

2) *Self-Organizing Maps (SOMs)*: Another popular clustering method is the SOM algorithm. This technique attempts to visualize similarity relations in a set of data items. Its main goal is to transform an incoming signal of any dimension into a one, or more commonly, two dimension discrete map. Because of this inherent property of SOM, it can also be viewed as a dimension reduction technique. Furthermore, since SOM implements an orderly mapping of a data of high dimension to a lower dimension, SOM can convert complex, non-linear relationships presents in the original data into simple geometric relationships in the lower plane [109].

A SOM consists of a grid of neurons, also known as prototype units, similarly to a NN. However, in SOM not only each neuron denotes a specific cluster learned during training, but neurons also have a specific location, so that units that are close to one another represent clusters with similar properties. To illustrate this concept, consider a SOM algorithm with a two-dimensional 4×4 grid, as shown in Fig. 7.

The way that SOM works is by having several units compete for the current input. Once a sample is fed into the system the SOM network determines which neuron the current sample is closest to by measuring the weight between the current sample and all possible neurons. The neuron that has the closest weight, usually measured by a distance metric like the Euclidean distance, then is the winning node, commonly known as the Best Matching Unit (BMU) and the sample is then assigned to that cluster.

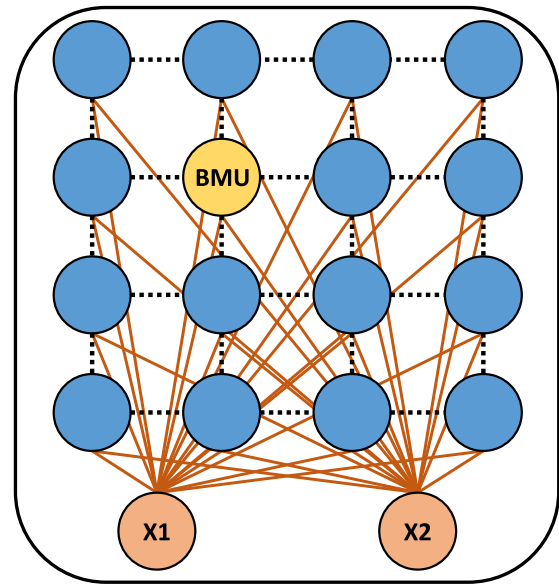


Fig. 7. An example of a 4×4 SOM network. The input layer, shown in orange, consisting of a two dimensional vector fully connected to all nodes of the SOM network. When an input is fed into the system, the weights between that sample and all possible clusters are measured. The neuron that has the closest weight is then assigned as the winning neuron (BMU), shown in yellow, and the input sample is classified as belonging to that cluster.

In cellular networks, SOM can be applied in the configuration of operational parameters [83], coverage and capacity optimization [110], HO management [89], [90], resource optimization [83], fault detection [93]–[96], [108], [111], and cell outage management [52].

3) *Anomaly Detectors*: Another group of algorithms that is quite popular nowadays are the ones involving Anomaly Detection (AD) techniques. These techniques have as main goal to identify data points that do not conform to a certain pattern observed in the data. These points are known as anomalous and typically mean that something is wrong or, at least, different than the usual behavior of a system.

There are several types of anomaly detection algorithms, they can be supervised, semi-supervised and unsupervised, but by far, the most common type found in SON applications is the unsupervised version. However, these unsupervised anomaly detection algorithms can be very different from each other. Some algorithms rely on the measurement of statistics from the initial data and measuring how far new data points are from the initial distribution. Other techniques rely on the density surrounding a set of points and based on how dense this region is the new point is then labeled as normal or anomalous. On the other hand, other algorithms can depend on the measurement of correlation between new points and the trained data or even on deviations from a simple set of rules [112], [113].

For readers interested in anomaly detection approaches focused in wired communication networks, a good resource is [114]. In cellular systems, anomaly detection algorithms are used mainly in self-healing to detect abnormal network behavior [97]–[102], [108], fault classification [98], [99], and perform cell outage management [9], [49], [50], [52], [70], [71], [104], [105], [107], [115], [116].

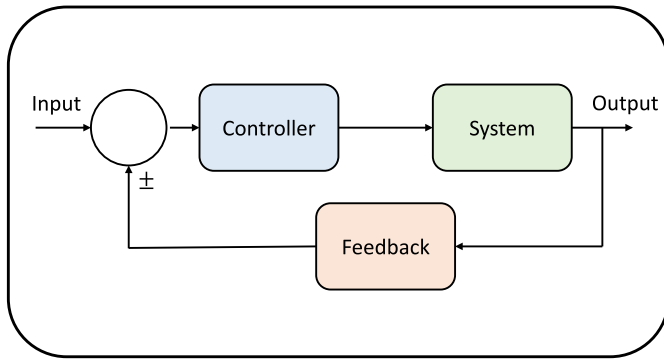


Fig. 8. Block diagram of a Feedback Controller. The controller takes actions, which affect the system. Then, based on the output response of the system, a feedback signal is produced and compared with the desired input response. After this comparison, this error signal is fed back into the controller.

C. Controllers

Although controllers do not belong to the class of intelligent algorithms, they have been extensively used to perform basic SON tasks in cellular networks due to their simplicity and ease of implementation. There are several types of controllers, but the most commonly used in cellular applications are the closed-loop controllers, where the output has an influence over the inputs (feedback controllers), and fuzzy logic controllers. Below is a description of closed-loop and fuzzy logic controllers together with some examples in the context of cellular systems.

1) *Closed-Loop Controllers*: Also known as *Feedback Controllers*, rely on a feedback mechanism between the input and output in order to constantly adjust its parameters.

Closed-Loop controllers have as primary objective to maintain a prescribed relationship between the input and output. These systems are able to do that by comparing the input-output function and measuring the difference between the ideal relationship (a rule that is embedded in the controller) and the current function to control the system. Figure 8, shows a simple diagram of a feedback controller. By measuring this difference (also called the error), the controller parameters can be tuned and the desired performance can be achieved. For more on closed-loop controllers, please see [117].

These controllers can be found in all domains of SON, and its applications include but are not limited to: NCL configuration [118]–[120], radio parameters configuration [121]–[123], coverage and capacity optimization [122]–[130], HO optimization [131]–[144], load balancing [145], [146], resource optimization [123], [147]–[149], coordination of SON functions [150]–[152], fault detection [153] and cell outage management [154]–[161].

However, since closed-loop controllers change their parameters only based on the error measurement between the current output-input function and the desired one, they are not as robust as other techniques that apply more sophisticated and intelligent methods. Nonetheless, this category of algorithm is the most researched and applied category of all references cited in this paper, as it can be seen from the previously given examples.

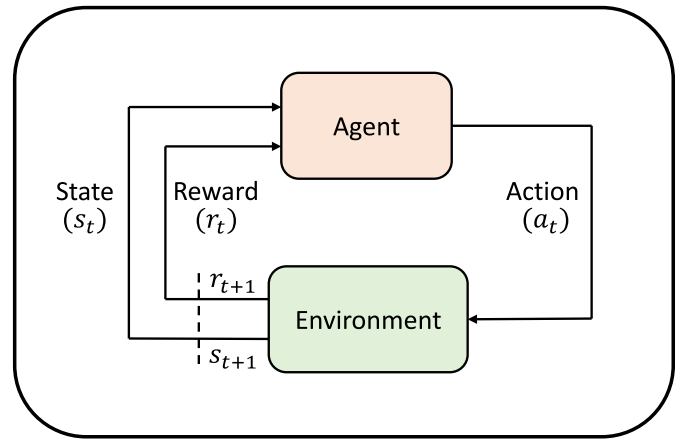


Fig. 9. Block diagram of a RL system. The agent takes actions based on its current state and the environment it is inserted. The agent also receives a reward or penalty depending on the outcome of its actions.

2) *Fuzzy Logic Controllers*: Another very popular type of controller in cellular systems applications is the Fuzzy Logic Controller (FLC). In contrast to normal feedback controllers, that use classical logic (Boolean logic), these controllers use fuzzy logic, a type of logic that represents partial truths. This process is done by applying an interpolation between the two extremes of binary logic (0 and 1). Since these controllers have a better granularity than standard binary logic controllers, generally, more detailed and complex solutions can be achieved by FLCs than feedback controllers.

A typical fuzzy controller has three main phases: fuzzifier, inference engine and defuzzifier. The purpose of the fuzzifier is to translate the current inputs of the system to fuzzy logic language. Normally these inputs are translated into linguistic terms, such as: *very low*, *low*, *normal*, *high* and *very high*, for example. After that, the inference engine applies a set of rules that will define the mapping between the input and outputs of the system. Lastly, the defuzzifier produces a quantifiable result by aggregating all the rules. For more information on fuzzy logic and fuzzy controllers, please see [162] and [163].

In terms of applications in cellular systems, fuzzy controllers are applied in self-optimization and self-healing problems, such as backhaul optimization [164], HO optimization [7], [90], [165]–[173], load balancing [173]–[175], resource optimization [7], [176]–[182] and fault detection [183].

D. Reinforcement Learning (RL)

Another learning technique quite popular is RL. This learning method is based on the idea of a system, in this context named as an *agent*, that interacts with its surroundings, senses its current state and the state of the environment, and chooses an action.

However, what differentiates an RL system from others is the process that comes after the action was taken. Depending on the action and its consequences, the agent can receive either a *reward* if the action taken was good, or a *penalty*, if the action was bad [184]. Figure 9 shows a basic diagram of RL.

Typically a RL system is divided into four stages.

- 1) Policies, which are responsible for mapping states into actions taken by the agent;
- 2) Reward function, which provides an evaluation of the current state and gives a reward or penalty depending on the results of the action taken previously;
- 3) Value function, which evaluates the expected reward from the chosen state in the future, given the possibility of an agent to evaluate a state in the long-term;
- 4) Environment model,² which determines the states and possible actions that can be taken by the agent;

Because of this reward mechanism that the RL algorithms have, there is also this trade-off notion of expectation versus exploitation, in which the agent must decide if it is better to explore what result taking another action would have in the system (exploration), or if it is better to keep the current knowledge and maximize the rewards of the current known actions (exploitation).

The most popular RL algorithms are: Q-Learning (QL), which uses Q-functions to find the best policies of the system and Q-Learning combined with Fuzzy Logic, also known as Fuzzy Q-Learning (FQL). For readers interested in Reinforcement Learning, please refer to: [184] and [185].

In cellular systems, RL algorithms are quite popular and are applied mainly in self-optimization. Some of its applications include: radio parameters configuration (self-configuration) [186]–[188], caching [189], back-haul optimization [190]–[193], coverage and capacity optimization [186]–[188], [194], [195], HO parameters optimization [196]–[198], load balancing [175], [199]–[201], resource optimization [193], [202]–[207] and cell outage management (self-healing) [49], [71], [186]–[188], [208], [209].

E. Markov Models

These stochastic models are mainly used in randomly changing systems and must obey the Markov property. The Markov property is a well-known property in statistics and refers to the memoryless property of a stochastic process. It states that the conditional probability distribution of future states depends only on the value of the current state and it is independent of all previous values [53], [210]. There are several different Markov models, but the most common ones applied to cellular networks are Markov Chains (MC) and Hidden Markov Models (HMM). The main difference between MC and HMM is the observability of the system states. If the states are fully visible, then MCs are the best option, else, if the states are partially visible or not visible at all, HMMs are preferred. Figure 10 shows a typical discrete-time Markov Chain model.

In the context of cellular systems, Markov models are mainly applied to self-optimization and self-healing. Applications include: mobility management [211]–[215], resource optimization [216], [217], fault detection [218] and cell outage management [219].

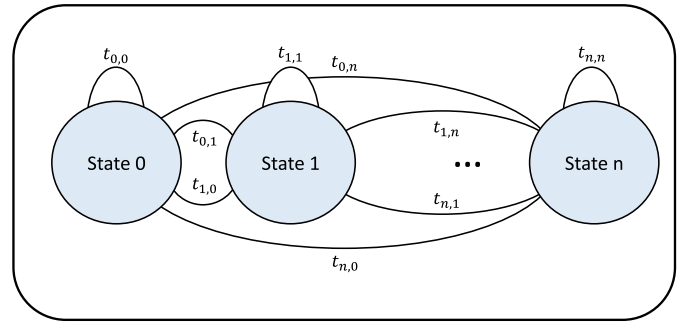


Fig. 10. An example of a Discrete-time Markov Chain, in which states are represented by circles, and transition probabilities between states are assigned by $t_{a,b}$.

F. Heuristic Algorithms

Heuristic algorithms basically consist of simple algorithms that follow certain guidelines or rules in order to make the best possible decision for the system at a given time. Normally these algorithms are applied when there is no known solution to a specific problem, or the solution is too costly to compute. By using heuristics, an approximate and sub-optimal solution can be found.

A simple example of heuristic is brute-force search, which is used when solutions to problems are impractical to be calculated. Another class of heuristic methods is the metaheuristics. Similarly to basic heuristic methods, metaheuristics also follow a set of basic rules, but in contrast to the prior approach, metaheuristics are more complex and more high-level, which lead to more optimized solutions than simple heuristics. For more information on heuristics, please follow: [220].

In the context of cellular systems, heuristics are applied mainly in self-optimization in coverage and capacity optimization [221], [222], and load balancing [223]–[226].

Another commonly found type of heuristics are the Genetic Algorithms (GA), which were inspired by concepts from nature, such as evolution and natural selection. As its nature counterpart, GAs use the mechanism of evolution and survival of the fittest in order to evolve a family of solutions and find the best solution after a certain number of generations. Further reading on GAs can be found on [227]–[229].

Despite being quite simple, GAs can not only find solutions to complex problems, but also solve non-deterministic problems. In the context of cellular networks, these algorithms can be found applied to solve all different kinds of issues, from radio parameters configuration [230], coverage and capacity optimization [231]–[233], HO optimization [234], [235], load balancing [236], resource optimization [57], to cell outage management [237]–[239].

G. Dimension Reduction

Dimension reduction can take two forms, feature selection or feature extraction. Feature selection consists of algorithms that select only the best, or most useful, features from an initial set of features. On the other hand, feature extraction algorithms rely on transformations applied to the initial set of features in order to produce more useful and less redundant attributes.

²Sometimes the environment model is not available or it is too difficult to simulate, hence, this stage is optional.

The main motivation behind dimensionality reduction techniques is to reduce the complexity of classifiers. In addition to complexity reduction, these techniques are also used to improve performance of algorithms and provide better generalization, as they aim to remove redundancy and less useful data from the initial data set [240].

In SON applications, the most popular dimension reduction techniques are Principal Component Analysis (PCA) [42], [107], [183], Minor Component Analysis (MCA) [42], [103], Diffusion Maps (DM) [39], [241] and MultiDimensional Scaling (MDS) [9], [49], [50], [70], [71].

All of these techniques apply a certain kind of transformation to the original data set in order to convert it to another space. PCA and MCA, for example, apply an orthogonal transformation in order to maximize the variance of the variables in the transformed space. MDS, on the other hand, tries to reduce the dimension of the original data set such that the distance between the items in the transformed space reflects the proximity in the original data.

Lastly, DM is a non-linear technique that tries to reduce the dimension of the data by analyzing geometrical parameters of the data set. In other words, the DM technique analyzes the position between points in the original data set, which can be measured by the Euclidean distance, and tries to produce a reduced version in which the diffusion distance in the transformed space matches the original Euclidean distance.

For more on dimensionality reduction techniques, please refer to [242]–[244].

H. Transfer Learning

Basically, Transfer Learning (TL) consists of applying a known model used in a previously known data set to another application. Despite seeming quite unintuitive, this knowledge transfer between different domains can provide significant improvements in learning performance, as no new model needs to be trained. TL can be applied in regression, classification and clustering problems and it has no restriction on the type of ML technique used. For further reading on TL, please refer to [245].

In cellular systems, TL can be found in caching [246], resource optimization [205], and fault classification [247].

III. LEARNING IN SELF-CONFIGURATION

Self-configuration can be defined as the process of automatically configuring all parameters of network equipment, such as BSs, relay stations and femtocells. In addition, self-configuration can also be deployed after the network is already operable. This may happen whenever a new BS is added to the system or if the network is recovering from a fault and needs to reconfigure its parameters [10].

In [6], for example, Wainio and Seppänen propose a generic framework in order to tackle the problem of self-configuration, self-optimization, and self-healing. From the perspective of self-configuration, the authors provide some basic steps that are needed to achieve an autonomous deployment of the network. The steps are as follows: first, the authors assert that a

BS should already have its basic operational parameters configured before being deployed, so that no professional skilled persons are required to deploy it. After that, the second stage consists of scanning and determining the BS's neighbors and creating a NCL. Lastly, the new deployed BS configures its remaining parameters and the network adjusts the topology in order to accommodate it. Other authors, such as in [223], propose a solution based on an assisted approach, in which after the deployment of a new BS, it senses and chooses a neighbor and request it to download all the necessary operational parameters. After that, the BS configures its remaining parameters automatically.

Regardless of the approach taken, it can be seen that both solutions have a few steps in common. These steps can be divided into:

- 1) Configuration of operational parameters;
- 2) Determination of new BS neighbors and creation of NCL;
- 3) Configuration of the remaining radio related parameters and adjustment of network topology;

In order to perform self-configuration, several learning techniques are being applied in order to configure, not only basic operational parameters, but also to discover BSs neighbors and perform an initial configuration of radio parameters. However, due to the increasingly complexity of BSs, which are expected to have thousands of different parameters that can be configured (many with dependencies between each other) and the possibility of new BSs joining the network or existing ones failing and disappearing from their neighbors' lists, the process of self-configuration still provides quite a challenge for researchers.

Based on these steps, three major use cases of self-configuration can be defined and are reviewed below, together with their ML solutions.

A. Operational Parameters Configuration

The first stage of self-configuration consists of the basic configuration of a BS, in which it learns its parameters so that it can become operable. These parameters can be IP address, access GateWay (aGW), Cell Identity (CID), and Physical Cell Identity (PCI). In addition to these parameters, other authors, such as in [83] and [248], also propose to perform network planning in an autonomous way.

Imran *et al.* [248] propose a framework to characterize the main Key Performance Indicators (KPI) in a LTE cellular system. After that, the authors' hybrid approach, which combines holistic planning with a semi-analytic model, is used in order to formulate a multi-objective optimization problem and determine the best cell planning parameters, such as: BSs location, number of sectors, antenna heights, antenna azimuth, antenna tilts, transmission power and frequency reuse factor.

On the other hand, Binzer and Landstorfer [83] develop a SOM solution in order to optimize the network parameters of a Code Division Multiple Access (CDMA) network. The solution optimizes not only planning parameters, such as the number of BSs in a certain area and their location, but also

radio parameters, like an antenna's maximum transmit power and its beam pattern.

Regarding the configuration of basic parameters, several works have been proposed, such as: [6], [84], and [223]. Wainio and Seppänen [6] develop a last-hop backhaul oriented solution, which offers solutions in all realms of SON, covering self-configuration, self-optimization and self-healing.

Hu *et al.* [223] propose a self-configuring assisted solution for the deployment of a new BS without a dedicated backhaul interface for LTE networks. According to the authors, first, the new BS should get the IP addresses of itself and the Operation, Administration and Maintenance (OAM) center. This can be done via Dynamic Host Configuration Protocol (DHCP), BOOTstrap Protocol (BOOTP) or by multi-cast by using the Internet Group Management Protocol (IGMP). After that, the new BS searches nearby neighbors and connect with one of them in order to request and download the remaining operational and radio parameters.

In terms of intelligence, the solutions presented in both [6] and [223] are not very adaptive as they require either a pre-configuration of its parameters or the assistance of other BSs. By its turn, the approach presented in [84] proposes the self-configuration of PCI and coverage related parameters in a heterogeneous LTE-A network scenario. In terms of PCI configuration, a grouping-based algorithm, that divides PCI resources and BSs into clusters and segments them into subgroups, is proposed. After that, each site is assigned into a specific subgroup where the domain BS is assigned with the first PCI and others BSs with random PCI of the same subgroup. By monitoring the PCI used by other BSs, the algorithm allows the network to maximize the PCI reuse distance and as a result it can avoid multiplexing interference effectively.

B. Neighbor Cell List (NCL) Configuration

Another important configuration parameter of BSs is the NCL. Whenever a new BS is added to the system, it must sense and discover its nearest neighbors in order to connect to them, so that basic network functions, such as HO, can be enabled. Two different tasks must be performed by an autonomous NCL algorithm. First, it must discover the neighbors of a newly deployed BS and, secondly, it must make the new BS known to its neighbors so that it can be added to their lists.

However, most of the research in literature focuses on the former [118]–[120], [144], [249], with the exception of, for instance, [6], which focuses on the latter. Furthermore, most of these solutions rely on the use of feedback controllers in order to perform NCL configuration.

In solutions such as [119] and [120], the authors apply an automatic procedure of NCL configuration and update by ranking the neighbor cells of the newly deployed BS according to certain parameters, such as coverage overlap or number of HO. After this process is done, a list is built based on them and the neighbors are obtained. Other solutions, like [249], rely on an even simpler method, the use of a threshold. In their solution, the authors analyze if the SINR is higher than a certain threshold and, if that is the case, that neighbor is added to the NCL, otherwise it is discarded.

Lastly, Li and Jantti [118] build three different solutions, of varying complexity, in order to configure the NCL. The first solution is a pure distance based approach, which analyzes if BSs fall inside a circle of a given radius within the newly deployed BS and, if so, it adds them to the NCL. The second solution evaluates not only the distance but also antenna parameters of neighboring BSs and, based on cell overlap, it determines the NCL. Finally, the third algorithm evaluates neighbors based on their distance and antenna parameters. However, differently than the previous solutions, where the radius was fixed, this time the authors calculate the optimal distance based on transmission power and using the Okumura-Hata path loss model.

Another approach that does not involve feedback controllers is [6], in which authors propose a solution for the new BS to be added to existing NCL. Their approach requires the existing BSs to scan the environment periodically using a beaconing mechanism, in which the BSs would exchange information between themselves and the new BS could be integrated into the existing network.

C. Radio Parameters Configuration

After NCL configuration, the BSs must configure its remaining radio parameters in order to become fully operable and provide service. The configuration of these parameters can involve the adjustment of transmission power, antenna azimuth and down-tilt angles, pilot transmission power, HO parameters (like hysteresis and Time To Trigger - TTT), and topology reconfiguration (backhaul configuration).

In [6], for example, Wainio and Seppänen propose a new backhaul update process, in which, after the newly deployed BS is configured, the network computes new routing paths and optimizes its topology in order to accommodate the new node. By reconfiguring the network backhaul and monitoring network resource utilization and performance, this solution can optimize the network's connections and provide better latency, reliability and energy saving.

Other techniques, such as [84], [250], and [251] aim to adjust the radio parameters based on measurements and data gathered from its neighbors. On [250], for example, Sanneck *et al.* propose a framework for self-configuration of a LTE BS, in which a subset of BS parameters was assigned dynamically. The solution proposed, Dynamic Radio Configuration Function (DRCF), assesses the coverage area of its neighbors in order to determine the best parameters of the new BS, form cell clusters and provide Tracking Area Codes (TAC) based on neighboring cells. Similarly, Eisenblatter *et al.* [251] build an antenna down-tilt and transmit power configuration mechanism based on its neighbors. The authors first state that the new BS should be deployed with low power and high down-tilt settings and as the new BS communicates with its neighbors, it would slowly adjust these values.

Another work that leverages the use of data in order to optimize its parameters is [84]. In this work, the authors propose a mechanism to adjust transmit power levels in order to mitigate interference between neighboring cells.

TABLE II
SELF-CONFIGURATION USE CASES IN TERMS OF MACHINE LEARNING TECHNIQUES

		Operational Parameters	NCL	Radio Parameters
Unsupervised Learning	SOM	[83]	-	-
	Misc.	[84]	-	-
Controllers	Feedback	-	[6], [118]–[120]	[6], [121]–[123]
Reinforcement Learning	FQL	-	-	[186]–[188]
Heuristics	GA	-	-	[230]
Miscellaneous	-	[6], [223], [248]	[249]	[84], [250], [251]

Another solution that can be encountered in radio parameters configuration is the feedback controller, as it can be seen from [121]–[123]. In [121], for example, Mwanje *et al.* build an algorithm for self-configuration of HO parameters, mainly hysteresis and TTT, in a LTE network scenario. To determine the best HO parameters, the authors define a HandOver Aggregate Performance (HOAP) metric, which depends on the RLF Rate, HO rate, and Ping-Pong rate. The algorithm then searches for an optimal point by adjusting the variables (one at a time) after a certain period of time and also depending on the feedback of previous HOAP measurements.

On the other hand, Claussen *et al.* [122] propose two algorithms that automatically adjust pilot power of a femtocell. The first algorithm is purely distance based, in which the femtocell power is configured so that at its edge, it has the same power of the strongest macrocell BS. The second solution uses the same principle as before, but it measures the received macrocell power instead of estimating it. By comparing the macrocell power at certain time intervals and due to the variations in the wireless channel, the power of femtocells can be constantly adjusted in a feedback loop.

Zhao and Chen [123] apply a self-configuration scheme for femtocells which improves indoor coverage and promotes energy efficiency of the network. Similarly to [122], the algorithm for self-configuration is distance-based and works by adjusting the transmit power of each femtocell to a value that is on average equal to the strongest power received from the strongest macrocell at a radius of 10 meters. By constantly adjusting this power, the authors are able to achieve a constant cell range for the femtocell.

Another learning technique that is quite popular is RL, more specifically, FQL, as it can be seen from [186]–[188]. In [188], for example, Razavi *et al.* propose the configuration of antenna down-tilt in order to adjust its coverage and capacity. The authors analyze their distributed algorithm in a LTE network scenario and present three different learning strategies, comparing them in terms of learning speed and convergence properties. The three different strategies are in terms of how many cells of the network can execute the FQL algorithm at the same time. In the first case, the authors test only one cell per time slot, in the second scenario the authors allow all cells to update at the same time and in the third scenario a mid-term approach is proposed, in which cells are divided into clusters and only one cluster is allowed to update its down-tilt

angle per time slot. Results show that all approaches are able to learn optimal antenna down-tilt angle settings, but the first and second approach can be either too slow or too complex, respectively. Hence, the authors conclude that the best solution, that provides a good trade-off in terms of speed and complexity, is the third one.

Similarly to [188], Razavi *et al.* [187] propose a distributed FQL algorithm in order to configure antenna's down-tilts in a LTE network scenario. The authors evaluate their algorithm performance in terms of spectral efficiency and also compare their algorithm with a related fuzzy algorithm, the Evolutionary Learning of Fuzzy rules (ELF). Another solution that utilizes the concept of FQL is the work in [186]. In this paper, the authors attempt to change an antenna's down-tilt angle setting in order to achieve self-configuration, self-optimization and self-healing in LTE networks. The authors compare their solution with the standard ELF solution and also consider two sources of noise, thermal and receiver noise.

Another paper that proposes a solution to self-deploying and self-configuring networks is [230]. In this work, the authors apply a GA solution to automatically configure BSs pilot transmit power levels, while also enabling the reconfiguration of their powers whenever a BS is added or removed from the network. Upon deployment, the BSs would enter a state in which they would seek surrounding neighbors and approximate their distances by adjusting its power levels accordingly. After this process is done, the BSs keep updating themselves by using feedback measurements from mobile users in order to make minor adjustments to cell sizes and fill possible gaps that might exist in the network.

A summary of the self-configuration use cases and their respective learning techniques is presented in Table II.

IV. LEARNING IN SELF-OPTIMIZATION

In SON, the concept of self-optimization can be defined as a function that constantly monitors the network parameters and its environment and updates its parameters accordingly in order to guarantee that the network performs as efficiently as possible [10]. Since the environment in which the network is inserted is not static, changes might occur and the BSs might need to adjust its parameters in order to accommodate the demands of the users. Changes can be in terms of traffic variations, due to an event happening in a certain part of a

city for example, coverage, due to a network failure, capacity, because of a change in users mobility patterns, such as a road block or an accident, and many others.

Due to this fact, some of the initial parameters configured in the self-configuration phase might not be suitable anymore and can require a change in order to optimize the network's performance. Since there are lots of different optimization parameters in the network, many ML algorithms can be applied. In addition, mobile operators also collect lots of data during network operation, which further enables the application of intelligent solutions in order to optimize the network. However, despite the huge amount of data collected, self-optimization is still a challenging task, as many parameters have dependencies between them and a change in one of them can alter operation of the network as a whole.

Based on the use cases defined by [12] and the literature reviewed in this paper, SON use cases in terms of self-optimization can be defined and will be described in the following.

A. Backhaul

One important aspect of future cellular network systems is the backhaul connection, or in other terms, the connection between the BSs and the rest of the network. Current cellular systems only evaluate the quality of the connection between the end-user and the BS. In the future, however, as systems will require to support a wider range of applications and different types of data, this approach might not be suitable and a more end-to-end approach, considering the whole link between the user and the core network might be better. With that in mind, some researchers developed solutions in order to solve the backhaul problem in future networks in terms of QoS and QoE provisioning [190]–[193], congestion management [6], [252] and also topology management [164].

Solutions such as [6] and [252] propose a backhaul solution involving flexible QoS schemes, congestion control mechanisms, load balancing and management features. In these solutions, the authors demonstrate a test-bed involving a network consisting of twenty nodes and with separated control and data plane. Another possible solution for backhaul optimization is proposed in [164], in which the authors utilize a FLC to arrange the network topology in response to changes in traffic demand.

Other backhaul optimization solutions are the works proposed by Jaber *et al.* [190]–[193]. In these works the authors used QL to intelligently associate users with different requirements, in terms of capacity, latency and resilience, to small cells depending on the backhaul connection that they offered. If the backhaul and the user needs match, then the user would be allocated to that cell, otherwise a new cell is searched. Results showed that the proposed solutions were able to achieve better QoE for all users at the cost of a small decrease in total throughput.

As it can be seen, the concept of backhaul optimization, despite being very promising and also considered a necessity for future networks, is not that popular, hence, future research directions can point to this area.

B. Caching

During the last couple of years, the fast proliferation of smart-phones and the rising popularity of multimedia and streaming services led to an exponential growth in multimedia traffic, which has very stringent requirements in terms of data rate and latency. In order to address these requirements and also reduce network load, specially during peak hours, future cellular networks must be coupled with caching functions. Some problems that arise, however, are the decision of what, where and how to cache, in order to maximize the hit-ratio of the cached content and provide gains to the network.

Wang *et al.* [253] provide a good overview of why caching is necessary in future networks, what might be the gains of caching at different locations within the network and also presents some of the current challenges encountered. In terms of caching solutions, several approaches are being considered, such as in [17], [82], [85], [86], [189], and [246].

Zheng *et al.* [17] explore various ways of integrating big data analytic with network resource optimization and caching deployment. The authors propose a big data-driven framework, which involves the collection, storage and analysis of the data and apply it to two different case studies. The paper concludes that big data can bring several benefits in mobile networks, despite of some issues and challenges that still need to be resolved.

Other caching solutions, like in [82], analyze the role of proactive caching in mobile networks. In this paper, the authors analyze and propose two solutions. First, the authors develop a solution to alleviate backhaul congestion. This mechanism caches files during off-peak periods based on popularity and correlations among users and file patterns and is based on the concept of CF. The second solution analyzes a scenario that explores the social structure of the network and tries to cache content in the most relevant users, allowing a Device-to-Device (D2D) communication. These influential users, as they are called, would then have content cached into their devices and disseminate it to other nearby users. By using *K*-means algorithm, this second approach can cluster users and determine the set of influential users and which users can connect to them.

Another approach from the same authors as in [82] is shown in [246]. In this work the authors apply a new mechanism based on TL in order to overcome the problems of data sparsity and cold-start problems that can be encountered in CF. In this new solution, the authors assume that they have gathered data and built a model for a source domain, composed of a D2D based network. After that, the proposed TL solution smartly borrows social behaviors from the source domain to better learn the target domain and builds a model that can smartly cache contents into the BSs. A figure showing this process is shown in Fig. 11.

Other solutions for caching optimization include the work in [85] and [86], where the caching problem is modeled as a game theory problem. Hamidouche *et al.* [85] model the system as a many-to-many matching game and propose an algorithm that is capable of storing a set of videos at BSs in order to reduce delay and backhaul load. On the other hand, Blasco and Gündüz [86], tackle the optimization problem of

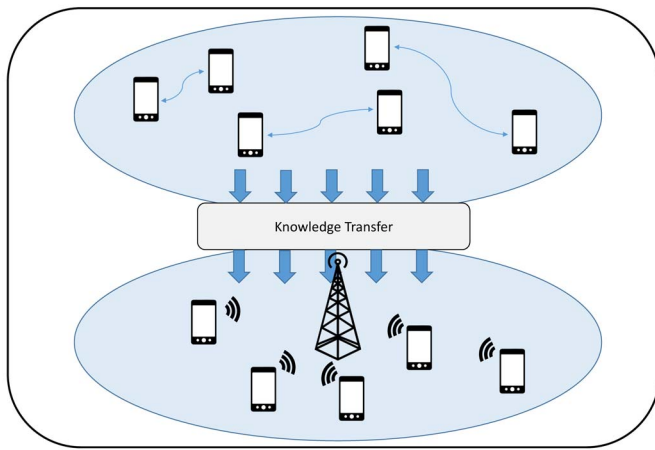


Fig. 11. An illustration of the solution in [246] based on TL. The system consists of two domains, on the top, the source domain, composed of a network based on D2D connections. On the bottom, the target domain, which considers a normal scenario of a BS (with limited backhaul link capacity) serving users. After data is gathered from the source domain and a model is built, it can be transferred to the target domain.

storing the most popular contents in order to relieve backhaul resources.

Another work that researched the impact of caching in mobile networks is [189]. In this solution the authors propose the optimization of caching in small cell networks and divide it into two sub-problems. First, a clustering algorithm (spectral clustering) was utilized in order to group users with similar content preferences. After that, RL is applied so that the BSs can learn which contents to cache and optimize their caching decisions.

C. Coverage and Capacity

Another challenging issue in future network systems is the optimization of coverage and capacity, in which the network tries to optimize itself in order to achieve the best trade-off between coverage and capacity. Based on this, several authors are proposing intelligent solutions to tackle this problem.

In [110], for example, Debono and Buhagiar apply SOM to optimize the number of cells inside a cluster and also antenna parameters in order to achieve a better coverage. In this work, the authors propose two different scenarios. The first scenario changes only cluster sizes, while the second one changes both cluster sizes and antenna parameters. On top of that, two SOMs are considered to perform cluster optimization. It is shown that the first scenario provides a gain of around 5%, while the second one achieves a gain in the order of 13%.

Other approaches, such as in [122], [126], and [127], utilize feedback controllers in order to optimize the coverage and capacity of the network. Claussen, *et al.* [122], develop a coverage adaptation mechanism for femtocell deployments that utilizes information about mobility events of passing-by and indoor users to optimize femtocell coverage.

Fagen *et al.* [126], propose a method to simultaneously maximize coverage while minimizing the interference for a desired level of coverage overlap. This optimization can be done for individual BS, a cluster of BSs or the whole network. On

the other hand, Engels *et al.* [127], develop an algorithm that tunes transmit power and antenna down-tilt angle in order to optimize the trade-off between coverage and capacity via a traffic-light based controller.

Furthermore, the work in [222] considers a novel Multi-Objective Optimization (MOO) model and proposes a meta-heuristic approach in order to perform coverage optimization. The solution simulated a LTE network scenario and aimed to maximize the performance of users in a given cell in terms of fairness and throughput. Other solutions, such as in [232] and [233], attempt to optimize the coverage of femtocells by using GAs. In both solutions, the authors tried to perform a multi-objective evaluation and the algorithm would try to satisfy three rules simultaneously: minimize coverage holes, perform load balancing and minimize pilot channel transmit power. In the end, the solution returns the best individual of all populations and changes the pilot power of femtocells accordingly.

1) *Antenna Parameters*: Another set of parameters that also have an impact on coverage and capacity of the network are the antenna parameters, mainly: antenna down-tilt and azimuth angles, and transmit power. In particular, the optimization of the antenna parameters often requires tuning after the initial operator's configuration and are very delicate, requiring not only an expert, but also a lot of precision to perform. Hence, it can be quite costly for the operators to perform this optimization and that is why several papers are trying to automatically optimize the antenna's parameters.

Joyce and Zhang [124] propose four different methods in order to optimize traffic offload of macrocells to microcells. The first two solutions utilize only microcell measurements, while the third method is based on Minimization of Drive Test (MDT) measurements and the last method is a hybrid of all three previous solutions. All methods, however, aim to maximize capacity offload from macrocells, or in other terms, maximize microcells' coverage. By changing the antenna down-tilts and transmission powers according to the measurements collected via a feedback loop mechanism this offload is achieved.

Gerdenitsch *et al.* [125] develop an optimization algorithm to find the best settings for antenna down-tilt angle and common pilot channel power of BSs. The solution begins by performing an evaluation of the network and analyzing the obtained results. After that, an iterative process formed by a control loop begins. In this process, parameters are changed according to certain rules and how far the parameters are from optimal until an accepted level is reached.

Other works, such as in [186]–[188] aim to optimize the down-tilt angle of the antennas by applying FQL in a LTE network scenario in order to achieve better coverage. While in [221], Eckhardt *et al.*, propose an algorithm for antenna down-tilt angle optimization in order to optimize the spectral efficiency of users. The approach considered a LTE network scenario and is based on heuristics to find the best antenna parameters.

2) *Interference Control*: Interference has always been a problem affecting the performance of communications systems and in future networks this will not be different. Hence, several

intelligent approaches are being considered in order to cope and control this limiting factor.

In [128], for example, Yun and Shin propose a distributed self-organizing femtocell management architecture in order to mitigate the interference between femtocells and macrocells. The solution consists of three feedback controllers, in which the first loop controls the maximum transmit power of femtocell users, the second determines each femtocell user's target SINR and the third attempts to protect the users uplink communication.

Another approach that involves the application of feedback controllers is the work in [129]. In this work a distributed algorithm applied to LTE networks, which performs Inter-Cell Interference Coordination (ICIC), is proposed. The algorithm assigns resources to cells and works similar to a frequency planning solution. It consists of two phases: in the initial phase, each cell attempts to assign resources by itself and, in the second phase, cells optimize themselves by resolving sub-optimal assignment of the resources. It is shown that the algorithm is capable of achieving good results and also assign resources reliably.

Mehta *et al.* [130], develop two solutions in order to address the problem of co-layer interference (interference between neighbors) in a heterogeneous macrocell and femtocell network scenario. The two schemes attempt to mitigate co-layer interference while also improving the minimum data rate achieved by femtocell users and ensuring fairness to them. The first scheme proposes a modification to the technique of Adaptive Frequency Reuse (AFR) by adding power control techniques to it, while the second scheme applies a self-organized resource allocation solution based on a feedback controller in order to allocate resources and manage the interference.

Zhao and Chen [123] also build a self-configuration and optimization scheme for a network of femtocells overlaid on top of a macrocell network. The algorithm automatically configures the femtocells transmit power and promotes self-optimization via a feedback controller to automatically control when to turn on or off femtocells in order to reduce interference between macro and femtocells.

Other approach to interference mitigation is the work in [194]. In this work, the authors model the coexistence of a macrocell and femtocell network and develop a distributed algorithm for femtocells to mitigate their interference towards the macrocell network. The authors divide the problem into two sub-problems of carrier and power allocation. The carrier allocation problem is solved via QL, in which at every time instant every femto-BS is in a given state. The femto-BSs then build their local interference map in every carrier, take an action and receive an immediate reward. While the second sub-problem, of power allocation, is solved using a gradient method.

Another solution that utilizes the concept of RL, is the work in [195]. In this paper Dirani *et al.* propose a solution to the problem of ICIC in the downlink of cellular Orthogonal Frequency-Division Multiple Access (OFDMA) systems. The problem is posed as a cooperative multi-agent control problem and its solution consists of a Fuzzy Inference System (FIS),

which later is optimized using QL. The solution is based on the concept of adaptive soft frequency reuse and the ICIC concept is presented as a control process that maps system states into control actions, which can be modeled as a RL system. The authors consider that the state of the system is defined by its transmit power, mean spectral efficiency and aggregated spectral efficiency, the available actions consist of reducing the transmit power by a certain amount and the reward is defined as the harmonic mean of the throughput.

Lastly, another solution comes from Aliu *et al.* [231]. In this work the authors adopt a novel Fraction Frequency Reuse (FFR) based on GA for ICIC in OFDMA systems. The main difference of this solution is that it not only attempts to use a new technique, but also considers a non-uniform distribution of users and characterizes it by determining its center of gravity. The proposed solution aims to, first, find the center of gravity of each sector and current state of each sector and then apply a GA to obtain the global state of all sectors.

D. Mobility Management

Another important aspect of future cellular network systems is the ability to predict user's movement in order to better manage resources and reduce the cost of network functions, such as HO. Mobility management can be defined as the process in which the network is able to identify in which cell the user currently is [59]. Current location techniques involve databases that store the locations of the users and every time the user changes position these databases need to be updated [27]. As it can be seen, this method is not very efficient. If the network could predict a user's next cell or even the path it will traverse, several gains in the network performance could be observed, hence, different solutions are being developed to this challenging problem.

Some papers, such as in [27], [28], [44], and [59]–[63] use back-propagation NNs in order to predict the next cell a user can be. The basic idea behind all these papers is to use the concept of NN to learn a mobility-based model for every user in the network and then make predictions of which cell the user is most likely to be next.

In [60], for example, Premchaisawatt and Ruangchaijatupon develop a method consisting of two cascaded ML models. The first model performs clustering via *K*-means while the second does classification. In classification, the authors compare the performance of three different methods, mainly, NN, DT and naive Bayes. Results show that the proposed model achieves better accuracy than performing only classification alone and also that the classifier that performed the best was the DT classifier. Despite using NNs as primary intelligent strategies, some papers also use different learning techniques. Akoush and Sameh [44] combine the concept of NN with Bayesian learning in order to perform classification tasks and predict a user's next cell and show that Bayesian networks outperform standard NN by 8% to 30%.

Another supervised technique that can be found in the mobility use-case is SVM. In [30], for example, Chen *et al.* build a model that uses only Channel State Information (CSI) and HO history to determine a user's mobility pattern. Their

algorithm defines an user trajectory based on the previous and next cell it traversed and, given the input data (previous cell and CSI sequence), the next cell can be predicted. In addition, the solution considers multiple classifiers, one for each possible previous cell, and trains several non-linear SVM classifiers with Gaussian kernels. On the other hand, Feng and Chang [68] consider the problem of estimating not only the location of mobile nodes in an indoor wireless network, but also channel noise. The solution uses a Hierarchical SVM model, composed of four different levels and is able to maintain good accuracy for speeds up to 10m/s.

Other approaches to mobility prediction are the works in [211] and [212], in which the authors propose a movement prediction and a resource reservation algorithm, which uses MC and HMM, respectively. Mohamed *et al.* [211] considered a discrete-time MC in order to represent cell transitions and determine a user's path. This approach does not require any training and optimization is done online. For each HO that happened, a transition matrix is updated and next predictions are made. Results show that the proposed solution is able to correctly predict a user's trajectory depending on a confidence parameter and also to reduce signaling cost of the network. On the other hand, the solution of [212], models the network as a state-transition graph and converts the problem into a stochastic problem. HMM is then applied, so that it learns the mobility parameters and, later, makes its predictions. Another solution that relies on the use of MC is the work in [213], in which Fazio *et al.* propose a movement prediction and a resource reservation algorithm. The movement prediction algorithm is done via distributed MC while bandwidth management is done in a statistical way.

In another set of solutions, this time from Sas *et al.* [254], [255], the problem of users that have high mobility and experience frequent HOs is addressed. The algorithm shown in [254] consists of three major components, a trajectory classifier, trajectory identifier and a traffic steerer. The objective of the algorithm is to classify and match current trajectory of users with previous trajectories stored in a database. After that, the steerer is activated so it can decide if it is better to keep the user in the current cell or to perform a HO. The solution in [255] builds upon that and adds a mobility classifier module before the steerer makes a decision. By implementing this classifier, the algorithm becomes more generic and can determine in which categories users fall into, e.g.,: slow, medium or high mobility, before deciding if they need to be steered or not.

Yu *et al.* [29] proposed a novel approach based on activity patterns for location prediction. Instead of predicting directly a user's next location, the solution attempts to, first, infer what the user's next activity is going to be, to, later, predict the location. The approach consists of three phases. The first phase tries to infer the current activity that the user is doing, the second attempts to infer the next activity and the third predicts the location. The proposed algorithm uses a supervised model to build an activity transition probability graph, which also takes into account the variation of time, so at different times of the day, the activities predicted by the model might be different, as it should be. To predict a user's next activity and location,

the paper uses the Google Places Application Programming Interface (API) which maps places to activities and determine a set of location candidates. Based on the result of the model, the location that has the highest probability is then chosen as the most probable location. Results show that this model is more robust than others and is also capable of achieving a higher accuracy on early stages than others methods.

The work proposed in [91] attempts to use semi-supervised or unsupervised techniques to reduce the effort of gathering labeled data to perform location prediction. To perform this, the authors build a discrete model and assign a Gaussian distribution to model the signal strengths of received signals by users for every location. After that, two different approaches are taken. In the first approach, the authors label only part of the data, making it a semi-supervised model, while in the second approach a data set with no label is considered. After that, the authors learn a model and use it to compute the estimate of location for each test sample. The authors conclude that there is significant opportunity to explore semi-supervised and unsupervised learning techniques since even without any labeled data, a reasonably accuracy could be obtained.

Recent work by Farooq and Imran [214] propose the use of a semi-Markov model together with participatory sensing in order to predict mobility pattern of users in the network. Another recent work, is the work proposed by Mohamed *et al.* [215], in which the authors build upon the previous model presented in [211]. By using an enhanced MC to predict next cell locations for users of the network, the authors demonstrate that by predicting a user's next location HO signaling costs can be reduced.

E. Handover Parameters Optimization

The process of changing the channel (frequency, time slot, spreading code or a combination of them) associated with a connection while a call is in progress is known as HandOver (HO). HO are of extreme importance in cellular networks due to the nature of mobility of its users. Without this procedure, mobility could not be supported as connections would not survive the process of changing cells. HO can be divided into two categories, there can be Horizontal HO, in which a user switches between BSs of the same network or Vertical HO (VHO), in which a user switches between BSs of different networks.

The optimization of HO parameters is crucial in many aspects of the network, as it can affect not only the mobility aspect, but can also affect coverage, capacity, load balancing, interference management, and energy consumption to name a few. Furthermore, the tuning of HO parameters also has an influence in several other metrics used by operators which are important to determine if the network is performing well, such as ping-pong rate, call dropping probability, call blocking probability, and early or late handovers [142].

Due to its importance, a substantial amount of research is being done in this area and several ML approaches are being considered. In [84], for example, Peng *et al.* discuss the impact that changing the A3-offset, and Time To Trigger (TTT) parameters or the application of certain techniques, such as

mobility estimation or Cell Range Extension (CRE) can have in the HO procedure. The authors also propose a solution for the Mobility Robustness Optimization (MRO) case and demonstrate the performance gains of CRE in a heterogeneous network scenario. Other authors, such as Soldani *et al.* [143], propose a generic framework for self-optimization and evaluate the impact of pruning NCL in terms of HO.

One possible solution to the optimization of HO parameters, can be in terms of NN, as seen in [35], [36], and [64]. In [35], for example, Narasimhan and Cox develop a new HO algorithm based on probabilistic NNs and compare it with the current hysteresis method. Results show that the NN reduces the number of HOs performed, reducing the cost of signaling of the whole network. On the other hand, Bhattacharya *et al.* [36] and Ekpenyong *et al.* [64], propose algorithms to optimize the HO procedure and better determine when an user needs a HO.

Another technique utilized in order to optimize the HO procedure is SOM. Sinclair *et al.* [89] develop a method to optimize two HO parameters, hysteresis and TTT, and achieve a balance between unnecessary HOs and call drop rate. The proposed algorithm has a different view from the main solutions, as it is more interested in which cell to tune the parameters, rather than how to tune them. Also, their model is based on a modified version of SOM, XSOM, which allows for a kernel method to replace the distance measurements of SOM, allowing a non-linear mapping of inputs to a higher dimensional space. Results show that the XSOM solution is able to reduce the number of dropped calls and unnecessary HOs by up to 70%.

On the other hand, Stoyanova and Mahonen [90], propose two different methods to solve VHO optimization. The first method is based on a FLC and involves measuring certain metrics, like: signal strength, bit error rate, latency and data rate in order to vote pro or against the HO for each mobile terminal. While the second approach involves the use of SOM, in which a few parameters (same as previous method) are periodically measured and, each of them, independently, can cause a HO initiation. Results show that the fuzzy solution performs really well and allows a simultaneous evaluation of different HO criteria. Unfortunately, the same cannot be said for the SOM solution. The authors conclude that SOM might not be appropriate for HO decision-making.

Another class of algorithm that is widely used in HO optimization is the class of feedback controllers, as can be seen from [131]–[142], and [144]. All of these solutions aim to change HO parameters, such as hysteresis, TTT, A3-offset, HO margins, cell offsets or stability periods based on the measure of performance metrics and how far they are from optimal.

FLCs are also widely used in the context of HO optimization, as it can be seen in the works of [7] and [165]–[172]. All of these algorithms consists of gathering certain network related metrics, fuzzifying them and making decisions in order to optimize HO margins, thresholds, hysteresis, TTT, or other attributes, so that the network can make better HO decisions.

Other solutions proposed for the optimization of HO parameters are in the context of RL. Mwanje and Mitschele-Thiel [196], develop a distributed QL

solution for the MRO use case. The contribution of the paper lies on the fact that their solution, QMRO, is able to adjust HO settings (hysteresis and TTT) in response to mobility changes in the network. Depending on the mobility observed in each cell, the algorithm applies a certain action and receives a penalty or reward. The solution in [197] also relies on QL. This time, however, the authors consider both MRO and Mobility Load Balancing (MLB) use cases. In the MRO solution, the primary goal is to determine optimal HO settings, while in MLB the objective is to redistribute load between cells.

Another solution to the HO optimization problem is the work of Quintero and Pierre [234]. In this paper, a hybrid GA solution is considered in order to solve the problem of assigning BSs to Radio Network Controllers (RNC) in a 3G network scenario. Another approach that uses GAs, is the work in [235], in which Capdevielle *et al.* propose a solution that enables every cell of a LTE network to adjust its HO parameters (HO margin, A3-offset and TTT), in order to minimize call drop and unnecessary HOs.

Bouali *et al.* [173] propose an algorithm based on a FLC combined with a fuzzy multiple attribute decision making methodology in order to choose which network should a user connect to, depending on the users' application and its requirements. Furthermore, results show that the proposed scheme is also capable of performing load balancing.

Another solution to HO management is proposed in [65], in which the authors utilize two NNs in order to determine which cell should an user handover to based on the user's perceived QoE in terms of successful downloads and average download time.

Dhahri and Ohtsuki [256] propose a cell selection method for a femtocell network. In this work three different approaches for cell selection are considered, first a distributed solution is proposed, secondly, a statistical solution is presented and the third solution relies on game theory. By determining which cell users should connect, the algorithm is able to maximize the capacity and minimize the number of HOs for every user of the network.

Another work also by Dhahri and Ohtsuki [198], proposes two different approaches for a cell selection mechanism in dense femtocell networks. The algorithms rely on QL and FQL and try to optimize, based on previous data, the best performing cell in the future for each user in the system. Results show that the enhanced FQL outperforms conventional QL and that the algorithms are capable of reducing the number of HOs while also maximizing capacity.

F. Load Balancing

In order to cope with the unequal distribution of traffic demand and to build a cost-efficient and flexible network, future networks are expected to balance its load intelligently. One solution, proposed in [34], aims to enable a heterogeneous LTE network to learn and adjust dynamically the CRE offsets of small cells according to traffic conditions and to balance the load between macro and femtocells. The algorithm utilizes a regression method in order to learn its parameters and then uses its model to adjust the CRE offsets.

Another approach involves the use of feedback controllers, such as in [145] and [146]. Viering *et al.* [145] build a mathematical framework to analyze network parameters and exemplify it on load balancing use cases. The algorithm attempts to modify HO thresholds in order to decrease the served area of overloaded cells and increase the area of underloaded cells and hence, achieve load balance. Similarly, Lobinger *et al.* [146] also develop a solution based on the control of HO parameters. This time, the goal is to find the best HO offset values between an overloaded cell and a possible target cell. Rodriguez *et al.* [174], on the other hand, propose the use of a fuzzy controller to achieve load balancing in LTE networks. The authors also implement a FLC in order to auto-tune the HO margins to balance traffic and reduce the number of calls blocked.

Munoz *et al.* [199] also propose the optimization of HO parameters to achieve load balancing by combining the concepts of FLC and QL in a 2G network scenario. Another similar work, is shown in [175]. This time, however, the authors investigate the potential of different load balancing techniques, by tuning either transmission powers or HO margins, to solve persistent congestion problems in LTE femtocells. The paper proposes solutions based only on FLC and also FLC combined with QL. Results show that the strategy that considered QL performed better and also performance gains were larger when QL was applied to optimize transmission power instead of HO margins.

Another approach that uses the concept of QL is the work by Mwanje and Mitschele-Thiel [200]. Their algorithm adjusts Cell Individual Offsets (CIO) between a source cell and all its neighbors by a fixed step and then applies QL in order to determine the best step value for every situation. The authors show that the new method performs better than a fixed-step solution. Another work that explores the QL concept is [201]. In this paper, Kudo and Ohtsuki build a scheme in which every user learns to which cell to send a service request in order to reduce the number of outages and also achieve load balancing.

Other solutions, such as in [223]–[225], attempt to solve the load balancing problem in a heuristic way. Hu *et al.* [223] develop an algorithm to balance unequal traffic load while also improving the system performance and minimizing the number of HOs. The algorithm relies in a greedy distributed solution and considers a LTE network scenario. Zimmermann *et al.* [224] propose a load balancing method by creating clusters dynamically via two different methods, centralized and decentralized heuristics. Lastly, the work of Al-Rawi [225], studies the impact of dynamically changing the range of low power nodes, by applying CRE. The solution aims to enable femtocells to take users from macrocells by adding a CRE offset to the received signal power of the users. Results show that dynamic CRE benefits the majority of users in the network, but does this by trading-off gains from picocell to macrocell users.

Du *et al.* [236] propose a dynamic sector tilting control scheme by using GAs to achieve load balancing. The solution aims to optimize sector antenna tilting to change both cell size and shape in order to maximize the system capacity. Another solution is the work in [226], in which an approach

is considered to balance load among neighboring cells of the network. The algorithm consists of five different parts in which it analyzes and determines which BS needs to have its traffic handled and determines to which neighbor to switch it to. The proposed method analyzes historical data collected by the algorithm, if available, and predicts which neighbor should have its antenna down-tilt angle changed and by how much. Otherwise, if no data is available, a heuristic search for the best neighbor is performed.

A recent work proposed by Basso *et al.* [92], present an unsupervised clustering algorithm in a control/data separation plane. Results show that the proposed solution is able to offload traffic from highly loaded cells to neighbor cells and that the algorithm can work in a high dense deployment scenario, making it suitable for future cellular networks.

G. Resource Optimization

Another important aspect of future networks is the optimization and provisioning of resources. One example is the work in [17], in which Zheng *et al.* explore various ways of integrating big data in the mobile network. In this paper, the authors propose a big data-driven framework and analyze use cases in terms of resource management, caching and QoE. All solutions are based on the collection and analysis of data in order to better determine how the network can change its parameters. The authors conclude by stating that big data can bring several benefits to future networks, however there are still significant challenges that need to be solved.

Some solutions, like the ones proposed in [31]–[33] and [55]–[58] rely on the use of NNs in order to optimize network resources. Sandhir and Mitchell [31] develop a scheme that predicts a cell demand after every 10 measurements taken by the system. At each prediction interval, the predicted resource usage in each cell is compared with the number of free channels available and channels are reallocated between cells, with the ones having more channels giving to the ones having less channels.

Another solution, proposed in [32], aims to predict user mobility by using two NN models in order to reserve resources in advance. Adeel *et al.* [33] build a cognitive engine that analyzes the throughput of mobile users and suggests the best radio parameters. The solution relies on the application of a random NN and three different learning strategies are investigated, Gradient Descent (GD), Adaptive Inertia Weight Particle Swarm Optimization (AIW-PSO) and Differential Evolution (DE). The authors show that AIW-PSO performs better and also converges faster.

Zang *et al.* [56] propose a method based on spatial-temporal information of traffic flow using *K*-means clustering, NN and wavelet decomposition to predict traffic volumes on a per cell basis and allocate resources accordingly. Another solution that applies NN, is the work in [55]. This time, however, the authors use a regression based NN and aim to predict the path loss of a radio link, in order to optimize the BSs transmission power. Another solution is shown by Railean *et al.* [57]. In this work, the authors develop an approach for traffic forecasting by combining stationary wavelet transforms, NN, and GA.

The paper adopts several different approaches based on similarity between days and also training of the NN and results show that when GAs were applied the performance decreased. Similarly, the work in [58], also develops a traffic forecasting solution and has as primary goal to determine voice traffic demand in the network.

Binzer and Landstorfer [83] builds a self-configuration mechanism that determines the number of BSs needed in the network and also a self-optimization technique in order to optimize BSs location and antenna parameters. From an optimization point of view, the algorithm relies in a SOM solution in order to move BSs accordingly and minimize the total number of under and oversupplied points in the network. The framework can also optimize the transmit power of BSs, antenna down-tilt angles and gains also using SOM.

Kumar *et al.* [87], propose a game-theoretic approach in order to optimize the usage of resource blocks in a LTE network scenario. The solution uses a harmonized QL concept and attempts to share resource blocks between BSs. Savazzi and Favalli [88], build two novel approaches for down-link spatial filtering based on *K*-means clustering algorithm. The first method groups users in clusters using *K*-means algorithm and then computes beam widths by considering the power level of edge users. The second method also uses *K*-means clustering, but after that, it compares for each user the best BSs available. Based on this, users might be reassigned to different BSs and overall system capacity can be increased.

Another approach is the work in [205]. In this work, an approach based on QL is investigated. The algorithm aims to adjust femtocells power, in order to maximize their capacity while maintaining interference levels within certain limits. In addition to QL, the paper also develops a TL solution between macrocells and femtocells, in which macrocells would communicate their future intended scheduling policies to femtocells. By doing this, the femtocells can reuse the expert knowledge already learned for a certain task and apply it to a future task.

Fan *et al.* [147] propose a cluster and feedback loop algorithm to perform bandwidth allocation. This algorithm explores user and network data in order to increase overall throughput. Kiran *et al.* [176] develop a Fuzzy controller combined with big data in order to find a solution for bandwidth allocation in RAN for LTE-A and 5G networks. On the other hand, Liakopoulos *et al.* [148] build an approach to improve network management based on distributed monitoring techniques. Their solution monitors specific parameters in each network BS and also considers that BSs interact with each other. Due to this interaction, BSs can take self-optimizing actions based on feedback controllers and improve network performance.

Dirani and Altman [202] propose a framework for Fractional Power Control (FPC) for uplink transmission of mobile users in a LTE network. The solution utilizes a FLC combined with QL in order to reduce blocking rate and file transfer times. Another solution that also utilizes QL is the work in [203]. In this paper, the authors develop a scheme to maximize resource utilization while constrained by call dropping and call blocking rates. Their solution can achieve performances comparable

to other classical methods, but has the advantage of not requiring explicit knowledge of state transition probabilities, like in Markov solutions.

1) *Call Admission Control (CAC)*: Call Admission Control (CAC) is a function of network systems that tries to manage how many calls there can be at a certain time in the system. Basically, if a new call comes to the network, either by someone making a new call or by transferring a call from another cell (via HO), this function determines if that call can be admitted or not in the system based on how many resources are available at that current time. Based on this, it can be said that CAC regulates access to the network and tries to find a balance between number of calls and the overall QoS provided, while also trying to minimize the number of dropped and blocked calls.

Several works have been published covering the optimization of CAC, such as [149], [177]–[182], [206], and [216]. In [149], for example, Lee *et al.* propose a CAC function that relies not only on information about the system resources, such as available bandwidth, but also on predictions made regarding system utilization and call dropping probability. By constantly monitoring these parameters and using a feedback controller, the authors are able to predict if a call should be accepted or rejected by the system for two different type of traffic classes, voice traffic and multimedia traffic.

Other authors, such as [177]–[182] rely on the use of FLCs in order to perform their CAC algorithm. Most of these solutions rely on estimating a set of parameters, such as effective bandwidth and mobility information in [177] and [181], cpu load in [178], or queue load in [179], to determine whether to accept or reject a call. On the other hand, the work in [216], propose a different approach to solve the CAC problem. In this work, the authors utilize a generic predictor scheme (in this case a Markov-based scheme) integrated with a threshold based statistical bandwidth multiplexing scheme in order to perform CAC for both active and passive requests. Based on the predictions given in terms of user mobility and time of arrival and permanence time, the algorithm then makes its decision.

Another approach to CAC is developed in [206], in which a RL solution is built in order to tackle the problem in a CDMA network. The solution involves four steps in order to work. First, data is collected and calls are either accepted or rejected based on any CAC scheme available. After that, the RL network is trained. The third step consists of applying the trained network to the simulated scenario and the fourth step consists of updating the network via a penalty/reward mechanism. Results show that the proposed method achieves better performance in terms of Grade of Service (GoS).

2) *Energy Efficiency*: Another problem that arises with the network densification process is the increase in energy consumption of the network. To overcome this issue, which would cut operators costs and also enable a greener network, several intelligent solutions are being developed.

One possible solution is proposed by Alsedairy *et al.* [7]. In this work, the authors introduce a network densification framework, however, instead of deploying regular small cells, the authors exploit the notion of cloud small cells and fuzzy logic.

These cloud cells are smart cells that underlay the coverage area of macro cells and, instead of being always on, they communicate with the macrocells to become available on demand. By optimizing the availability of small cells, the network can reduce its overall energy consumption.

Zhao and Chen [123] also propose a mechanism to promote energy efficiency in the network. Their solution relies on a feedback controller in order to determine when to turn on or off a femtocell. This is done by comparing the distance detected between an user and the femtocell and comparing its virtual cell size. The authors define the virtual cell size as a distance between an user and a femtocell, in which the SINR of the user is equal between macrocell and femtocell. By comparing this distance, the authors propose to turn the femtocells off if the distance exceeds the virtual cell size and to turn it on otherwise.

Kong *et al.* [204] build a scheme to dynamically activate or deactivate modular resources at a BS, depending on the network conditions, such as traffic or demand. The approach involves a RL algorithm, based on QL, that continuously adapt itself to the changes in network traffic and makes decisions of when to turn on an additional BS module, turn off an already activated module or to maintain the same condition. The proposed solution can achieve a very high energy saving, with gains of about 80% without increasing user blocking probability.

Peng and Wang [217] apply an adaptive mechanism to increase the standardized Energy Saving Mechanism (ESM) quality. The framework relies on adjusting sleep intervals of cells based on network load and traffic. The algorithm relies on the concepts of MC and can save network power and also guarantee spectral efficiency. The solution divides the energy saving process in three scenarios, heavy, medium and light loads, and, for each scenario, the adaptive solution is investigated. The authors conclude that the proposed adaptive solution is better than the standard solution, ESM, specially in light loads scenarios, while in higher loads, both schemes achieve similar performances.

Another solution is presented in [257]. In this work, the authors tackle the problems of improving traffic load and network planning. Their solution first builds supervised prediction models in order to predict traffic values and then applies the information gathered from external planned events in order to improve prediction quality. Based on the traffic demand prediction, the framework is then able to turn on or off certain cells in the network, achieving energy efficiency.

Recent work by Jaber *et al.* [193], tried to intelligently associate users with different BSs depending on their backhaul connections. In the proposed scenario, each BS had multiple backhaul connections and an energy optimization, in terms of which backhaul links to turn on and off, was performed.

Another recent solution is the work proposed in [207] by Miozzo *et al.*, in which QL was used in order to determine which BSs to turn on or off and to improve the energy usage of the network.

Lastly, the work in [258] utilizes big data, together with supervised learning (polynomial regression), in order to optimize the energy of ultra dense cellular networks. The authors

show that the proposed solution can achieve the highest cell throughput while maintaining energy efficiency, when compared to conventional approaches.

H. Coordination of SON Functions

Another important issue that arises with the advent of SON is how to coordinate and guarantee that two or more distinct functions will not interfere with each other and try to optimize or adjust the same parameters at the same time [73]. One simple example of this can be a hypothetical scenario where the network tries to minimize its interference level, but at the same time it tries to maximize its coverage. To avoid this type of situation, it is essential that SON functions are coordinated to ensure conflict-free operation and stability of the network.

Lateef *et al.* [73] develop a framework based on DT and policies in order to avoid conflicts related to the mobility functions of MLB and MRO. Also, another important contribution of the paper is that it classifies the possible SON conflicts into five main categories, mainly: KPI conflicts, parameter conflicts, network topology mutation, logical dependency conflicts and measurement conflicts.

Another approach that tries to resolve the SON conflict management is proposed in [150]. The authors consider a distributed coordination scenario between SON functions and analyze the case in a LTE network scenario. Each SON function can be viewed as a feedback loop and are modeled as stochastic processes. The authors were able to conclude that coordination is essential and that it can provide gains to the system.

Other solutions involving feedback controllers, can be seen in [151] and [152]. Lateef *et al.* [151] start by presenting a hybrid classification system of SON conflicts. The authors state that, since many SON conflicts can fall into more than one category, this hybrid approach is better and propose a fuzzy classification to accomplish that. The authors also evaluate some use-cases of SON conflicts and present distributed solutions based on feedback controllers, in which measurements are gathered, evaluated and the parameters changed accordingly.

Similarly, in [152], Karla also classifies SON parameters, but his classification is only based on the parameters impact on the cellular radio system, resulting in only two classes of parameters. Karla also presents a proof of concept scenario, in which a simplified LTE-A scenario is simulated and coordination is evaluated. First, the system performs a set of offline computations in order to find good configuration parameters and then the system uses a feedback controller to update itself in an online manner.

Table III shows a summary of the reviewed papers for the self-optimization use cases and how they are distributed in terms of ML techniques.

V. LEARNING IN SELF-HEALING

Current healing methods not only rely on manually interventions and inspection of cells, but also on reactive approaches, that is, the healing procedures are triggered only after a fault has occurred in the network, which degrades the network's

TABLE III
SELF-OPTIMIZATION USE CASES IN TERMS OF MACHINE LEARNING TECHNIQUES

		Backhaul	Caching	Coverage and Capacity	Mobility	Handover	Load Balancing	Resource Optim.	Coordination
Supervised Learning	Bayes	-	-	-	[44], [60]	-	-	-	-
	NN	-	-	-	[27], [28], [44], [59]–[63]	[35], [36], [64], [65]	-	[31]–[33], [55]–[58]	-
	SVM	-	-	-	[30], [68]	-	-	-	-
	DT	-	-	-	[60]	-	-	-	[73]
	CF	-	[82]	-	-	-	-	-	-
	Misc.	-	-	-	[29]	-	[34]	[257], [258]	-
Unsupervised Learning	K-Means	-	[82]	-	[60]	-	-	[56], [88]	-
	SOM	-	-	[110]	-	[89], [90]	-	[83]	-
	Game Theory	-	[85], [86]	-	-	[256]	-	[87]	-
	Misc.	-	-	-	[91]	-	[92]	-	-
Controllers	Feedback	-	-	[122]–[130]	-	[131]–[144]	[145], [146]	[123], [147]–[149]	[150]–[152]
	FLC	[164]	-	-	-	[7], [90], [165]–[173]	[173]–[175]	[7], [176]–[182]	-
Reinforcement Learning	QL	[190]– [193]	[189]	[194], [195]	-	[196]–[198]	[200], [201]	[193], [203]–[205], [207]	-
	FQL	-	-	[186]–[188]	-	[198]	[175], [199]	[202]	-
	Misc.	-	-	-	-	-	-	[206]	-
Markov	MC	-	-	-	[211], [213], [215]	[256]	-	[216], [217]	-
	HMM	-	-	-	[212]	-	-	-	-
	Misc.	-	-	-	[214]	-	-	-	-
Heuristics	Heuristics	-	-	[221], [222]	-	-	[223]–[226]	-	-
	GAs	-	-	[231]–[233]	-	[234], [235]	[236]	[57]	-
TL	-	-	[246]	-	-	-	-	[205]	-
Miscellaneous	-	[6], [252]	[17]	-	[254], [255]	-	-	[17]	-

overall performance and also results in a loss of revenue to operators.

The self-healing function in SON is expected not only to solve eventual failures that might occur, but also to perform fault detection, diagnosis and trigger automatically the corresponding compensation mechanisms. In addition, it is expected that future cellular systems also move from a reactive to a proactive scenario, in which faults and anomalies can be predicted and the necessary measures taken before something actually happens. Due to this change in paradigm in current cellular networks, self-healing solutions are extremely challenging and rely heavily on previous gathered data in order to build models and try to predict whenever a fault might occur in the network.

From a learning perspective, several ML algorithms can be applied, depending on the data that operators have and its nature. In some scenarios, it is easy to label certain types of data, such as in fault classification, in others, however, such as in outage cases, in which outage measurements appear to be normal or only deviate a slight amount from normal, it might be more suitable to not label the data and work with unsupervised algorithms.

In [259], for example, Barco *et al.* present a survey on state-of-the-art self-healing techniques and also propose a unified framework themselves. The paper defines a self-healing reference model, which would be composed of five core

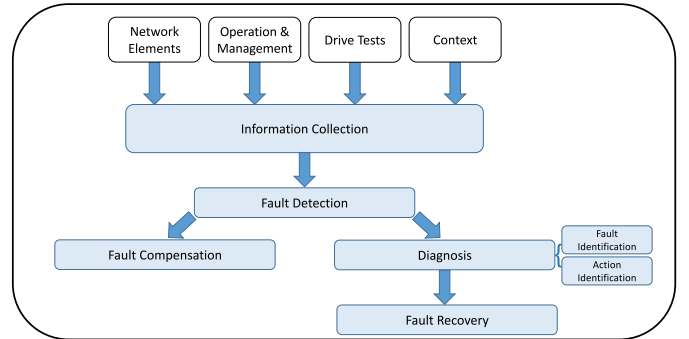


Fig. 12. Proposed self-healing reference framework. Adapted from [259].

functions: information collection, fault detection, diagnosis, fault recovery and fault compensation, as shown in Fig. 12.

Another example of a self-healing framework can be seen in [6]. In this paper, the authors show a reactive backhaul solution for 5G networks, which involves aspects of self-configuration, self-optimization and self-healing. From the self-healing point of view, the authors develop an event-based fault detection, in which a fault would always trigger a link state update message broadcasted from the point of failure. By combining a fast failure detection algorithm with offline computed paths, the authors show that the backhaul link can be recovered very quickly.

Based on the collected references and also from [11], which defines the major use cases for self-healing, the following use-cases for self-healing could be defined.

A. Fault Detection

The first and foremost thing a self-healing function must be able to do is to automatically detect when and where a fault occurred in the network. This can be done either by measuring certain KPI, estimating its values in the future, or even by trying to predict when a fault will occur in the network. Coluccia *et al.* [45], propose a solution based on Bayes' estimators in order to estimate the values of certain KPI and forecast when a failure might occur in a 3G network scenario. On the other hand, in [69], Ciocarlie *et al.* build an adaptive ensemble method to model and determine the performance status of cells in the network. The framework uses certain KPI to determine the state of a cell and uses a combination of different SVM classifiers in order to classify new observed data points.

Other papers, such as in [93]–[95], utilize a SOM algorithm in order to cluster and analyze cellular data. Raivio *et al.* [93], [94] show two classification methods based on SOM in order to monitor cell states and their performances in a 3G network scenario. Cells are clustered based on their performance levels and after that, each cell is classified according to certain categories, determining if its performance is acceptable or if it is degraded due to some fault. On the other hand, in [95], Kylvaja *et al.* model a solution that analyzes and identifies possible problematic cells in a 2G network. Another approach that involves the application of SOM is the work in [96]. In this paper, the authors build a mechanism to detect anomalies in the core network of cellular networks. First, the authors choose certain KPI to be monitored. After that, SOM is applied and anomalies are detected in terms of the distance between the weight vector of the BMU and the new state vector.

Other approaches, such as in [97]–[101], make use of statistical analysis and similarity-based methods in order to detect anomalies in the network. In [97], for example, Fiadino *et al.* build a framework to detect and diagnose anomalies via Domain Name System (DNS) traffic analysis. The algorithm monitors certain DNS features and as soon as one or more of them show a significant change, a flag is activated. Furthermore, the paper analyzes two different approaches, one relies on the entropy of the measured features, while the other is based on the statistical distribution of traffic. By comparing the two methods, the authors were able to determine that both solutions were able to detect short and long lived anomalies, but only the probabilistic solution captured the entirety of the long lived anomalies, while the entropy based approach detected only a slight deviation on the beginning of the event.

Szilágyi and Nováczki [98] develop an integrated framework for detection and diagnosis of anomalies in cellular networks. The detection is based on monitoring radio measurements and other KPIs and comparing them to their usual behavior while the diagnosis is based on reports of previous fault cases and learning their impact on different KPIs. Similarly, in [99], Nováczki builds a model, which improves the work presented before in [98]. The new framework has

the same objectives of performing detection and diagnosis of anomalies, this time, however, the authors build a new profile learning technique to classify the anomalies, which will be presented in the next section.

D'Alconzo *et al.* [100], propose a statistical-based anomaly detection algorithm for 3G cellular networks. The algorithm collects traffic data and identifies deviations in its distribution. By measuring the similarity between the measured distribution and the stored values it can detect and recognize when a fault happens in the network. Bae and Olariu [101] also utilize a similarity-based approach to detect anomalies. In their solution, a normal profile is built from normal mobility patterns of users in the network and then a dissimilarity metric is computed and evaluated to determine anomalies. Bouillard *et al.* [102], on the other hand, develop an online algorithm that uses the notion of constraint curves from Calculus and applies it to anomaly detection.

Tcholtchev and Chaparadza [153] approach the detection problem from a different point of view, from the operator's perspective. This work proposes considerations on how operational personnel can control automatic fault-management feedback loops and criteria that should be used for estimating whether a fault in the network should be reported to the operator or not. Liao and Stanczak [183] develop a novel framework based on dimension reduction and fuzzy classification in order to determine anomalies in the network. The proposed solution uses PCA to reduce the input's dimension and a kernel-based semi-supervised fuzzy clustering is employed to perform classification. By assigning samples to different classes and analyzing the trajectory of a sequence of samples, anomalies are predicted. Results show that the solution performs well in a LTE network scenario and is able to proactively detect anomalies associated with various fault classes.

Another work that tries to predict when a fault will occur in the network is the work in [218]. In this solution, a continuous time MC is utilized, together with exponential distributions, to model the reliability behavior of BSs in future cellular networks. The MC model is built with three states in mind: healthy, sub-optimal and outage cells and failures could be classified as trivial or critical. The paper analyzes three different case studies and, for each study, it tries to predict the occurrence of faults based upon past database values.

Hashmi *et al.* [108] compare five different unsupervised learning algorithms (*K*-means, Fuzzy *C*-means, SOM, local outlier factor, and local outlier probabilities) in order to detect faults in the network. Results show that SOM outperforms *K*-means and Fuzzy *C*-means.

Lastly, Gómez-Andrades *et al.* [111] propose to combine MDT measurements with SOM in order to detect whenever a fault happens in the network. The proposed solution was evaluated in two different LTE networks and demonstrated that it was able to diagnose and also locate (up to a certain degree) faults within the networks.

B. Fault Classification

Another important task that needs to be done whenever a fault occurs is its classification. This involves the

determination of the causes of the problem, so that the correct solution can be triggered. Nowadays, most methods rely on manual processes and are done by experts that need to diagnose and classify the problems. However, this is not optimal and can lead to misclassifications, leading to incorrect solutions and wasting operators time and money.

In [37], for example, Barco *et al.* present a system based on simple naive Bayesian classifiers in order to perform fault classification. The proposed framework focuses on troubleshooting RAN problems in 2G networks, but also addresses the issue of how the problems can be solved. The authors propose a three step approach. The first step identifies poor performing cells based on alarms and performance indicators. The second step finds the cause of the problem and the third step attempts to solve the problem by executing specific actions.

Another work that relies on the use of Bayesian techniques, is [38]. In this paper the authors build an automated diagnosis mechanism for 3G networks. The diagnostics system involves two components, a model and an inference method. The model is based on a naive Bayes classifier and, regarding inference methods, two were investigated: Percentile-Based Discretization (PBD) and Entropy Minimization Discretization (EMD). The authors compare both methods and determine that EMD performs better, so in the use-case proposed they just analyzed the system involving the naive Bayes and the EMD inference technique.

Puttonen *et al.* [74], apply a classification of RLF reports based on previously gathered information to identify coverage, HO or interference related problems. The classification is done using a DT and two use-cases are analyzed. The first use case considers a network with medium load, while the second case considers a high load scenario. The solution is efficient in terms of revealing the types of problems each cell can have and, thus, helping operators detect individual cell problems.

Other approach that performs fault classification is the work in [98] and [99], which uses anomaly detectors based on statistical analysis in order to diagnose faults in the network. Szilágyi and Nováczki [98] perform classification by comparing measured KPI values with reports of previous fault cases. While on [99], a new profile learning technique is proposed. This technique examines historical KPI data and identify its normal operational states. After that, it takes the current KPI values and analyzes its symptom patterns. By seeking the most similar pattern with the stored data a match can be done and the fault can be classified.

Another approach is presented in [247] by Wang *et al.* In this work the authors build a framework that relies on TL to diagnose problems in femtocells. The authors state that traditional diagnosis approaches are not applicable to femtocell networks because of the challenge of data scarcity. To overcome this issue, the authors utilize TL, so that historical data from other femtocells can be leveraged and used in order to troubleshoot problems. The authors also state that general TL techniques are not accurate, so they propose a new model, Cell-Aware Transfer (CAT). In this new scheme, two classifiers are trained and, after that, each classifier is treated as a voter in the diagnostic model. The final diagnosis is the result that gets the most votes. The authors compare their solution

with methods based on SVM and TL-SVM and show that CAT achieves higher accuracy than the other approaches.

C. Cell Outage Management

One of the SON use cases that has attracted a lot of attention recently is the automated detection of cells in outage condition. Self-healing solutions have to perform compensation mechanisms in order to overcome the outage scenario and minimize the disruption caused in the network. Current methods, however, involve manual detection of cell outages, which might take days, or even weeks, in order to be detected. With the increase in scale and complexity of future cellular networks, manual procedures will not be good enough and autonomous management, which involves detection and compensation, must be provided in SON.

Several researchers are trying to address the outage issue and provide intelligent solutions to this problem. One possible approach is shown in [40], in which Mueller *et al.* propose a cell outage detection algorithm based on NCL reports of mobile terminals. The algorithm attempts to use the NCL reports to create a graph of visibility relations between cells and, by monitoring the changes in this visibility graph, outage detection is performed. The authors also analyze three different classification techniques, involving a manually designed system, DT and linear discriminant analysis and show that the outage detection quality is largely based on the performance of the classification algorithm.

Feng *et al.* [41] attempt to classify cells into four different states, depending on the level of degradation in its performance: healthy, degraded, damaged and outage. The authors designed a back propagation NN with three layers and used a differential GA in order to train the model. Results show that the improved NN outperformed standard BP NN.

Onireti *et al.* [49] consider a network scenario which has distinct control and data planes and present a framework which is capable of detecting outages in both planes. In order to do that, the authors design two algorithms that monitor control and data cells. To perform cell outage detection, two approaches are taken. For control cells, two distinct algorithms are tested, K -NN and Local Outlier Factor based Anomaly Detector (LOFAD), while for data cells a heuristic approach was considered. On top of that, the authors also use MDS to perform dimension reduction in order to cope with the order of the input data. To perform compensation, the authors consider a RL approach, to adjust gains of antennas and transmit power in order to compensate the coverage and capacity degradation caused by the outaged cell. Results show that both control and data detection schemes are able to detect outages and that the K -NN algorithm outperforms LOFAD.

Xue *et al.* [51] also build a detection mechanism based on K -NN and a heterogeneous network scenario, consisting of macrocells and picocells. The model detects outage through cooperation between outaged cells (which were modeled as cells that had their performance degraded and were not completely out of service) and neighbor cells. The problem is then modeled as a binary classification problem and K -NN is implemented to classify the data. On the other hand, Zoha *et al.* [70]

present and evaluate an outage detection framework based on MDT reports. The framework aims to compare and evaluate the performance of two different algorithms: LOFAD and One Class Support Vector Machine based Detector (OCSVMD). The system is divided into two phases, profiling and detection. In the profiling phase, after collecting the MDT measurements and reducing the dimension of data, by applying MDS, the system builds a reference database based on the normal, fault-free, network scenario. After that, the two different models are applied in order to classify network measurements and determine cell outages.

Wang *et al.* [79]–[81] show a solution to outage management in femtocell networks. This time, however, it involves the application of CF. Wang *et al.* [79], [81] develop a detection mechanism involving two stages: triggering and cooperative stage. The triggering stage involves the application of CF, while the cooperative stage involves all femto-BS reporting to the macrocell BS to make a final decision. While in [80], Wang and Zhang analyze three different architectures for self-healing, mainly: centralized, decentralized and local cooperation and investigate their advantages, disadvantages and limitations. Also, under the local cooperation architecture, the paper builds an outage detection and compensation mechanism, similar to the previous discussed solution.

Another group of solutions for outage management consist on the analysis of statistics. In [104], for example, de-la Bandera *et al.* propose to detect outage via the analysis of HO statistics. Muñoz *et al.* [105] apply a solution that detects degraded cells through the analysis of time evolution metrics. The solution compares the measured metric with a generated hypothetical degraded pattern and, if they are sufficiently correlated, outage is detected. Lastly, Liao *et al.* [115], show an algorithm based in a weighted combination of three hypothesis tests to perform outage detection.

Another class of algorithms that is very popular in outage management are the feedback controllers. Most of the proposed approaches, such as in [154]–[159] and [161], aim to solve the problem of outage compensation by triggering certain mechanisms that will adjust coverage of neighbor cells and try to minimize the impact of the outaged cell in the system. Most solutions rely on the adjustment of transmission power, and antenna down-tilt angles. Other solutions, such as in [160] focused on the problem of outage detection in networks with separated control and data plane. Its goal is to detect outage in data cells and involves monitoring certain metrics and signaling outages whenever irregularities occurred. Figure 13, shows an example of outage management.

In [186]–[188], the authors aim to change the down-tilt of the antennas by applying FQL. Despite their solutions being primarily focused on self-configuration and self-optimization, the authors argue that, since the process of changing antenna parameters can be used in order to mitigate the effects of outage cells, their solutions could work from a self-healing point of view. Another paper that uses FQL is the work from Zoha *et al.* [71]. In this paper the authors develop a framework to address cell outage detection and compensation by using MDT measurement reports. Outage detection is done by first gathering the MDT measurements and reducing their

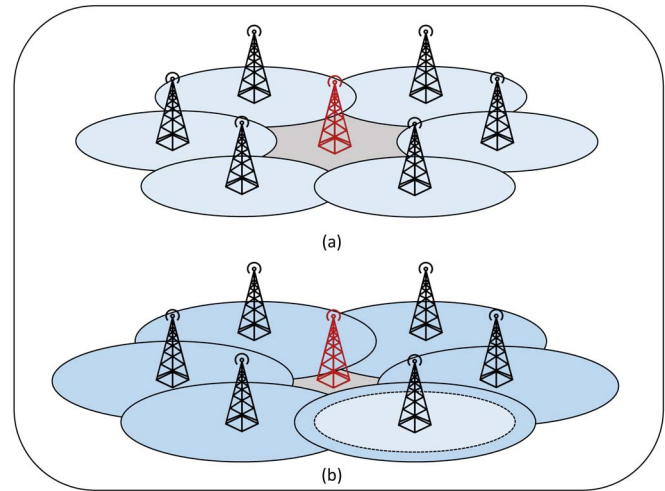


Fig. 13. An example of cell outage management. In (a), the network detects that the central site has suffered outage and triggers the appropriate self-healing mechanisms. Each neighbor cell, by triggering these mechanisms, adjusts their coverage area and, in turn, compensates for the outaged cell, providing service in the affected area, as seen in (b).

dimension using MDS. After that, the paper analyzes two different anomaly detection algorithms in order to detect the outage, LOFAD and OCSVMD. The compensation mechanism is based in a Fuzzy controller combined with RL in order to adjust antenna down-tilts and transmit powers and minimize the effects of the outaged cell.

Saeed *et al.* [208] also build a fuzzy controller combined with RL in order to perform cell outage compensation. The solution investigates three different methods, by adjusting only antenna down-tilt angle, only transmit power or both. On the other hand, Moysen and Giupponi [209] model a RL approach for cell outage compensation in LTE networks. Their solution aims to automatically adjust the transmit power and antenna down-tilt angles to provide coverage and capacity where needed.

Another possible solution for outage detection is [219]. In this work, the authors develop a solution based on HMM in order to classify BSs into four possible states: healthy, degraded, crippled or catatonic. In order for the system to estimate the BSs states, a set of measures reported by the users are collected and a state probability is produced accordingly. Results show that the proposed solution is able to predict a BS state with around 80% accuracy.

Other solutions, such as in [237]–[239], rely on the use of GAs in order to achieve outage management. Jiang *et al.* [237] propose a method based on immune algorithms in order to adjust the uplink target received power in surrounding cells so that both coverage and quality of the whole network can be maintained.

Li *et al.* [238] model a distributed architecture for cell outage management. This architecture consists of five phases and aims to solve quality and coverage problems caused by outages in LTE networks. In order to perform outage compensation, the algorithm increases the power of the reference signal with the objective of maximizing the coverage region while minimizing

coverage overlap. In order to determine the best parameters, a particle swarm, which is a type of GA, was implemented.

Ma *et al.* [106] propose an unsupervised clustering algorithm in order to tackle the problem of outage detection. In this work, the authors simulate two different outage scenarios, by reducing the antenna gains of two antennas by different amounts. Results show that the solution is able to classify differently the two outage scenarios and, hence, can enable mobile operators to choose appropriate compensation methods depending on the outage degree.

Lastly, a recent work by de-la Bandera *et al.* [116] proposes a method to analyze data and perform outage compensation based on either correlation or delta detection (threshold based).

1) *Sleeping Cells Management*: A particular type of problem that can occur in the network is the sleeping cell scenario. A sleeping cell is a special case of outage, which makes mobile service unavailable for users, but from an operator's perspective the cell still appears to be fully operable. Sleeping cells can roughly be classified into three groups: *impaired*, in which a cell is still able to carry traffic, but certain performance metrics are slightly lower than expected; *crippled*, in which a cell has severe degradation in its capacity and *catatonic*, in which a cell is completely out of service [42], [107].

Imran *et al.* [9] propose a case study in which the objective is to perform the detection of sleeping cells. Through network monitoring and observation, the authors create a model and use it to predict sleeping cells behavior. The model was based on K -NN-based anomaly detectors and also used MDS to perform dimension reduction.

Turkka *et al.* [39] build a data-mining framework that is able to detect sleeping cells, network outage and change of dominance areas. The main idea consists of finding similarities between periodical network measurements and previously known outage data. The solution, first gathers a set of MDT data and builds a reference database. After that, a new test database is created in order to classify the newly obtained samples. After this process, both sets go through a nonlinear DM process, which reduces their dimension and then data classification is performed via Nearest Neighbor Search (NNS), a supervised learning method similar to K -NN. Results show that because MDT data is used together with RLF events, a more reliable and faster detection is achieved.

Chernov *et al.* [42], also presents a data mining framework, but this work is focused in the detection of sleeping cells caused by Random Access CHannel (RACH) failure. Their algorithm collects user data, processes it and performs dimension reduction via PCA and MCA. After this process is done, the algorithm then performs two steps: first, it extracts outlier sub-calls from the data set by applying a K -NN anomaly detection algorithm. Then, the algorithm assigns sleeping cell scores to each cell, in which the higher the score, the higher the chance of a cell being in the sleeping state.

Zoha *et al.* [50] propose a solution in order to automatically detect sleeping cells. The model gathers MDT measurements from a normal network scenario, applies MDS to reduce the data's dimension and then learns its basic profile. After that, the authors propose two different solutions in order to detect sleeping cells, one is based on K -NN Anomaly Detection,

while the other is based on LOFAD. After the models predictions, the authors also perform sleeping cell localization, in order to classify which cell triggered the sleeping cell scenario. Results show that cells can be correctly localized and that K -NN outperforms LOFAD.

Another work regarding sleeping cell detection is the work of Chernov *et al.* [52] in which the authors analyze the detection of a sleeping cell due to a RACH failure, similar to [42]. In this paper, different anomaly detection algorithms are compared, such as K -NN, SOM, Local Sensitive Hashing (LSH) and Probabilistic Anomaly Detection (PAD). Results show that despite all algorithms being able to determine the sleeping cell condition correctly, the proposed solution has the best performance.

On the other hand, in [103], Chernov *et al.* develop a solution based on MDT reports and data mining techniques in order to detect sleeping cell conditions. In addition, the paper also considers that there are positioning errors associated with the MDT measurements. The authors first build a model based on a normal network scenario and then apply a certain anomaly detection algorithm to classify samples as anomalous or not. This time, the solution relies first in, reducing the data's dimension by applying MCA, and then applying the unsupervised technique of K -means to perform classification. Furthermore, since the authors also considered positioning error, the determination of which cell caused the sleeping cell condition is not trivial and three different methods to determine which cell is the sleeping cell are proposed. Another sleeping cell solution is shown in [107], in which Chernogorov *et al.* used PCA in order to perform dimension reduction and a Cluster Based Local Outlier Factor (CBLOF) to perform sleeping cell classification.

Lastly, another solution is proposed by Chernogorov *et al.* [241]. In this paper, the authors use DM, not as a dimension reduction technique, but rather as a classification tool in order to detect anomalies. The authors argue that DMs are able to convert non-linear data sets to linear in the new embedded space, so it could be used as a classification tool as well. After detecting the anomalies, the paper also develops a method to determine their locations, by determining the dominance map of every cell. Then, anomalies are mapped according to the dominance maps produced and the problematic cells can be identified.

Table IV presents a summary of the literature covered in the self-healing section in terms of the ML techniques utilized.

VI. ANALYSIS OF MACHINE LEARNING APPLIED IN SON

Intelligence in future networks is a promising concept, however, because each SON function has its own requirements, certain algorithms tend to work better for specific functions. In this section, the most common ML algorithms found in SON are compared in terms of certain metrics. These metrics relate not only to the performance of ML solutions, such as accuracy, amount of training data or convergence time, but also relate to the performance required for each *self-x* function, such as: scalability, complexity, and response time, for

TABLE IV
SUMMARY OF SELF-HEALING USE CASES IN TERMS OF MACHINE LEARNING TECHNIQUES

		Fault Detection	Fault Classification	Outage Management
Supervised Learning	Bayes	[45]	[37], [38]	-
	K-NN	-	-	[42], [49]–[52]
	NN	-	-	[41]
	SVM	[69]	-	[70], [71]
	DT	-	[74]	[40]
	CF	-	-	[79]–[81]
	Misc.	-	-	[39]
Unsupervised Learning	K-Means	[108]	-	[103]
	SOM	[93]–[96], [108], [111]	-	[52]
	Anomaly Detectors	[97]–[102], [108]	[98], [99]	[9], [49], [50], [52], [70], [71], [104], [105], [107], [115], [116]
	Misc.	[108]	-	[106]
Controllers	Feedback	[153]	-	[154]–[161]
	FLC	[183]	-	-
Reinforcement Learning	FQL	-	-	[71], [186]–[188], [208]
	Misc.	-	-	[49], [209]
Markov	MC	[218]	-	-
	HMM	-	-	[219]
Heuristics	GAs	-	-	[237]–[239]
Dimension Reduction	PCA/MCA	[183]	-	[42], [103], [107]
	DM	-	-	[39], [241]
	MDS	-	-	[9], [49], [50], [70], [71]
TL	-	-	[247]	-
Misc.	-	-	-	[40]

example. It is important to note that the classifications provided in this section are only general guidelines and based on overall performance of the considered ML methods.

A. Scalability

One important concept in ML algorithms is the notion of scalability. The scalability concept can be defined as an algorithm being able to handle an increase in its scale, such as feeding more data to the system, adding more features to the input data or adding more layers in a NN, without it limitlessly increasing its complexity [10].

In order to cope with future networks, which are expected to be much more dense and generate much more data, scalability is a highly desirable feature so that algorithms can be deployed easily and quickly in the network. Furthermore, the notion of scalability can also help in determining if certain types of algorithms can be mass deployed in decentralized solutions or if centralized solutions are preferred.

Examples of SON functions that require scalability can be in algorithms trying to predict mobility pattern of users in the network, as predicting the mobility pattern of a single user is very different than trying to predict pattern for all users of the network. Another example can be in the self-healing domain,

in which the whole network might be required to be monitored in order to detect and manage faults.

B. Training Time

Another important concept is the training time of each algorithm. This metric represents the amount of time that each algorithm takes to be fully trained and for it to be able to make its predictions.

Training of ML algorithms can be done either offline or online. Depending on the training that is carried out certain types of algorithms might be more suitable for certain SON functions. For example, functions that are heavily dependent on time, such as mobility management, HO optimization, coordination of SON functions or self-healing, would not be able to cope with algorithms that require high training times and perform online training, as they would not be able to generate a model, and consequently its predictions, in time for these applications. However, if the same algorithms can be applied with an offline training methodology, algorithms that were not suitable before can now fit into these more time restrict SON functions.

Examples can be the application of offline trained NNs to predict user mobility patterns, or the application of CF in order to perform outage management.

C. Response Time

Also related to the agility of a system is the response time of an algorithm. This metric represents the time that an algorithm takes, after it has been trained, to make a prediction for the desired SON function.

Contrary to the previous metric, in which algorithms that have high training times can still be applied to time sensitive SON functions if an offline training is performed, algorithms that have a high response time are not desirable for these SON functions, as predictions would not be generated in time.

SON functions, such as self-configuration do not require a fast response time, as most of the configuration parameters of a network can be determined in an offline manner, hence, algorithms that have a low response time can be adequate for these applications. Other types of functions, however, such as mobility management, HO optimization, CAC, coordination of SON functions and self-healing might require faster response times, leading to the application of faster algorithms.

D. Training Data

Also related to the parameters of ML algorithms is the amount and type of training data an algorithm needs. Algorithms that require lots of training data, usually have better accuracy, but they also take more time to be trained. Furthermore, as discussed before, certain types of algorithms only work with labeled or unlabeled data, which might fit best certain types of SON functions.

Algorithms that rely on high amounts of data to perform well, will also need more memory in order to accommodate the data and use it to train their models. This might not be compatible with certain SON functions, such as caching, or functions that need to be deployed at user terminals, such as mobility prediction, or HO optimization, as memory capabilities are limited. On the other hand, the huge amount of data collected by operators can also enable more complex and demanding solutions to be deployed in BSs, leading to an easier integration between SON and Big Data.

In the case of self-healing functions, for example, operators tend to gather lots of unlabeled data while the network is monitored. In this scenario the application of unsupervised or RL techniques might be more suitable to address these functions, while supervised techniques would not be applicable.

E. Complexity

Complexity of a system can be defined as the amount of mathematical operations that it performs in order to achieve a desired solution. Complexity also relates to the power consumption of a system, as a system that needs to perform more operations will, consequently, need more power to operate. Hence, this concept can determine if certain algorithms are more suitable to be deployed at the user or operator's side, for example. Furthermore, more complex systems also take longer to produce their results, however, when they do, these results tend to be better than other simpler approaches.

An example of highly complex algorithms are the GAs. By exploring all possible solutions, GAs are able to find near optimal solutions to a problem, but, usually, take lots of time

(generations) in order to reach these solutions. Simpler algorithms, such as Bayes classifiers or K -NN also have their merits, as being extremely simple facilitates the mass deployment of these algorithms and enable operators to have fairly decent results.

In terms of SON functions, usually, simpler solutions are preferred, however, sometimes simple solutions are not capable of providing sufficient results. In self-configuration, for example, as future networks are expected to be much more dense and BSs are expected to have thousands of parameters, simple solutions will not suffice and more complex solutions will need to be deployed. On the other hand, simpler solutions might fit self-healing functions, enabling future networks to become proactive and much quicker in detecting and mitigating faults.

F. Accuracy

Another important parameter of ML algorithms is their accuracy. Future networks are expected to be much more intelligent and quicker, enabling highly different types of applications and user requirements. Deploying algorithms that have high accuracy is critical to guarantee a good operability of certain SON functions. In caching optimization, for example, caching the right content, at the right place, at the right time is crucial in order to reduce the delay experienced by end users. Another example is in terms of fault detection, as correctly detecting faults in the network can lead to a quicker response by other SON functions and mitigate the impacts of faults in the network.

On the other hand, other types of functions might not require extremely high accuracy and can be more lenient regarding it. One example can be in the estimation of coverage area of a cell, in which the exact coverage area might not need to be determined, and an estimate can be enough. Another example can be in terms of load balancing, in which perfectly load balancing of the whole network might not be required, or even possible. Managing the load of the network up to a certain extend might be enough and more relaxed algorithms can be more suitable for these kind of applications.

G. Convergence Time

Another important parameter in which algorithms can be evaluated is their convergence time. Differently than the response time, which relates to the time an algorithm takes to make a prediction, the convergence time of an algorithm relates to how fast an algorithm agrees that the solution found for that particular problem is the optimal solution at that time.

Certain algorithms, such as controllers, or RL need an extra time to guarantee that their solution has converged and will not change abruptly in the next time slot. Since the convergence time adds an extra time in addition to the response time of a system, solutions that have this additional parameter might not perform well in time sensitive functions, such as mobility or HO optimization. However, by guaranteeing that their solution has converged and is the best solution possible for that time, this kind of algorithms can provide near optimal solutions to the system.

TABLE V
ANALYSIS OF MOST COMMON ML TECHNIQUES IN TERMS OF SON REQUIREMENTS

		Scalability	Training Time	Response Time	Training Data	Complexity	Accuracy	Convergence Time	Convergence Reliability
[53], [77], [260], [261]	Supervised Learning								
	Bayes	Low	Low	Low	Low	Low	Fair	-	Fair
	K-NN	Low	Low	Low	Low	Low	Fair	-	Fair
	NN	Fair	High	Low	High	High	High	-	Fair
	SVM	Fair	Fair	Low	High	High	High	-	Fair
	DT	High	Low	Low	Low	Low	Low	-	Fair
[53], [262]	CF	High	Fair	Low	High	High	High	-	High
	Unsupervised Learning								
	K-Means	High	High	Low	Low	Low	Fair	-	Low
	SOM	High	Fair	Low	High	Low	High	-	Fair
[117], [162]	Game Theory	Fair	-	Fair	-	Fair	Fair	-	Fair
	AD	Fair	Fair	Fair	High	Fair	Fair	-	Fair
[184], [185], [263]	Controllers								
	Feedback	High	-	Fair	-	Low	Low	Fair	Low
[264], [265]	FLC	Low	-	Fair	-	Fair	Fair	Fair	Low
	Reinforcement Learning								
[227], [229]	QL	Fair	High	Fair	Low	Low	Fair	High	Fair
	FQL	Low	High	Fair	Fair	Fair	Fair	High	Fair
[264], [265]	MC	High	Fair	Low	Fair	Low	Fair	-	Fair
	HMM	Fair	Fair	Low	Fair	Fair	Fair	-	Fair
[227], [229]	Heuristics								
	Heuristics	Low	-	Fair	-	Fair	Fair	-	Fair
[227], [229]	GAs	Low	High	High	High	High	High	-	High

SON functions that can benefit from this kind of algorithms can be, for example, functions in self-configuration, which are not time sensitive and need to carefully tune the initial parameters of a BS, caching optimization, and resource optimization.

H. Convergence Reliability

Another important parameter of learning algorithms is the initial conditions that they are set in and their convergence reliability. In this sense, this metric represents the susceptibility of an algorithm to be stuck at local minima and how can initial conditions affect its performance. Although related to accuracy, since algorithms that are able to minimize the impacts of being stuck at local minima can achieve more optimal solutions, this metric represents the susceptibility that an algorithm has in being stuck or not at local minima.

The majority of learning algorithms are susceptible to the local minimum problem, but by taking some actions this problem can be minimized. One possible action that can be taken is to initialize the algorithms with random small values, in order to break the symmetry and to reduce the chances of the algorithm being stuck at a local minima. Other types of actions that can be taken combined with this approach is to average the performance of the algorithm for different starting conditions or to provide a varying learning rate.

However, certain types of algorithms are able to produce solutions closer to the optimal, by exploring the whole

search-space, like in CF or GAs, which might be more suitable for functions that need reliability, such as self-configuration, caching and coordination of SON functions. Others algorithms, such as K -means or RL, can find different solutions to the same problem, which can be applicable to functions that do not require the best or a static solution to its problem, applications could be in the area of backhaul optimization, load balancing and resource optimization.

A more detailed view of how each of the most common ML algorithms found in the literature performs in terms of the aforementioned SON metrics is depicted in Table V. Furthermore, Table VI shows guidelines on when to utilize each ML algorithm for each SON use-case. Based on the requirements of each *self-x* functions, the performance of each algorithm for each SON metric, and the amount of references that utilize certain algorithm in that function, the authors were able to build Table VI, which provides general guidelines on when to use certain ML algorithms. It is important to note that Table VI serves only as a guidance and should not be strictly followed, as depending on the application and type of data available, different algorithms can be applied to different SON functions.

VII. FUTURE RESEARCH DIRECTIONS

In order for 5G to overcome the current limitations of LTE and LTE-A, it is clear that a shift in paradigms is needed and that different solutions to common problems need to be found.

TABLE VI
GENERAL GUIDELINES ON THE APPLICATION OF ML ALGORITHMS IN SON FUNCTIONS

		Self-Configuration					Self-Optimization					Self-Healing			
		Operational Parameters	NCL	Radio Parameters	Backhaul	Caching	Coverage and Capacity	Mobility	Handover	Load Balancing	Resource Optim.	Coordination	Fault Detection	Fault Classification	Outage Management
Supervised Learning	Bayes							✓	✓		✓			✓	✓
	K-NN		✓					✓	✓		✓		✓	✓	✓
	NN			✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
	SVM			✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
	DT	✓	✓	✓			✓		✓	✓	✓			✓	✓
	CF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Unsupervised Learning	K-Means		✓				✓	✓	✓	✓		✓			✓
	SOM	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓			✓
	Game Theory	✓	✓	✓		✓			✓	✓	✓				
	AD												✓		✓
Controllers	Feedback		✓	✓			✓		✓	✓	✓	✓			✓
	FLC		✓	✓	✓		✓		✓	✓	✓	✓			✓
Reinforcement Learning	QL	✓	✓	✓	✓	✓	✓		✓	✓	✓				✓
	FQL	✓	✓	✓	✓	✓	✓		✓	✓	✓				✓
Markov	MC						✓	✓	✓	✓	✓	✓	✓	✓	✓
	HMM						✓	✓	✓	✓	✓	✓	✓	✓	✓
Heuristics	Heuristics	✓	✓	✓	✓		✓			✓	✓	✓			✓
	GAs	✓	✓	✓	✓		✓			✓	✓				✓

However, despite current work being done in the area of SON, with an increase of maturity and robustness in the area, with more and more different ML algorithms being explored and applied in different contexts, there are still open issues and challenges that need to be addressed in order to enable a fully intelligent network in the near future. In the next sections, future research directions and open issues are explored and the role of ML algorithms in future cellular networks is also discussed.

A. Self-Configuration

This is the area with the least amount of research being done up to this moment. Nonetheless, interesting solutions can be found in the literature, which can lay down the foundation for future researchers in this area.

1) *Dense Environments*: One possible direction is the configuration of future networks in dense environments. As network densification is a critical component of future networks, it is essential to enable configuration in these kinds of networks. In the future, it is expected that several BSs will be deployed not only by operators, but by regular users, making it difficult for operators to track all BSs and to configure them manually. Hence, intelligent solutions need to be deployed and ML algorithms can create models that can enable the configuration of extremely dense and complex networks.

One example can be the application of GAs in order to configure a network and its topology. GAs, by exploring a large family of solutions, can perform these computations

offline and generate an optimal network model, prior to the deployment of the network.

2) *Non-Dense Environments*: Other currently unexplored scenario is the self-configuration of networks in rural or nomadic environments. Most of the reviewed papers focus on the self-configuration of networks deployed in dense urban environments. The deployment of cells in rural and not so dense environments could lead to different self-configuration solutions, since BSs do not need to be as densely deployed and capacity and coverage requirements are less stringent. One possible enabler for this scenario is the application of a scaled-down version of ML algorithms. For example, let's say that operators have trained their models in a dense network environment and what to apply the same model to a more sparse network. In this scenario, a scaled down version of the algorithm applied in the dense scenario can be deployed.

3) *NCL Configuration*: As it was demonstrated in the NCL section, most of the reviewed papers focused on the development of solutions that would enable the newly deployed BS to discover its neighbors. However, one aspect that is not that much explored is the fact that the new BS must make itself known to the other BSs in the network, so that their NCL can be updated as well. The development of solutions in this area would further enable the functionality of plug-n-play that is expected from future cellular networks as well as an autonomous reconfiguration of the network topology as new BSs are added into the system.

In order for BSs to learn their neighbors in a scenario where operators have no control of when new BSs are deployed,

intelligent solutions need to be applied. In this regard, one example can be the usage of clustering algorithms, in order to cluster nearby BSs, so that BSs that are within a cluster are all in each other NCLs, while BSs outside that cluster are not. Then, when a new BS is deployed, ML algorithms can then determine to which cluster the new BS belongs to and update all NCLs accordingly.

4) *Emergency Communication Networks*: Another area that is not very well covered in current literature is the reconfiguration of a network after a natural disaster occurs, in which the network is severely disrupted. Recent work by Wang *et al.* [261] surveys how big data analytic can be integrated to communication networks in order to understand disastrous scenarios and how data mining and analysis can enhance emergency communication networks.

In this regard, the application of ML solutions is essential to reestablish communications as fast as possible and to reconfigure the network with the remaining BSs. Since the network was already configured, the configuration of operational parameters of each BS is not needed, but a reconfiguration of its neighbors and radio parameters is necessary. ML solutions can help the remaining BSs to reconfigure themselves by automatically learning the new environment conditions and generating new models on the fly in order to reconfigure certain parameters, such as transmit power of each BS or antenna down-tilt, so that service can be restored as soon as possible.

B. Self-Optimization

Self-optimization together with self-healing are the areas that attract most researchers. Although several promising solutions have already been proposed in certain functions, such as in mobility management, HO optimization, load balancing, and resource optimization, there are still open issues that need to be addressed.

1) *Backhaul*: The optimization of backhaul connections between BSs and the core network is essential in future networks, however, as it can be seen, there is not much work covering the backhaul optimization process. Since in the future a huge amount of users is expected to access the mobile network with different types of applications at the same time, the management of backhaul resources is of extreme importance. In this sense, ML algorithms can be deployed in order to learn individual user patterns and requirements, based on the applications that each user is using and learn to which backhaul should users connect to. Furthermore, another possible research area can be in the investigation of cells with different backhaul solutions. In this scenario, ML algorithms can determine which and how much each backhaul should be utilized in order to optimize energy consumption of the network while also attending user needs.

2) *Caching*: Caching is essential in future cellular networks in order to enable low latency to end-users and to provide a better QoE. However, several issues regarding how, what and when to cache are still persistent and need to be investigated [253]. In order to address these issues, the analysis of user behaviors, such as which contents are more popular, and at what time of the day, can be a key enabler in

caching solutions. By analyzing different user behaviors, ML algorithms can then be applied and different models can be created in order to decide which contents to cache and at which BS.

3) *Coordination Between SON Functions*: Another interesting area of research could be on the management and the interoperability of different SON solutions. Since SON functions rely on the autonomous change of network parameters in order to adjust its settings, coordination between these functions is of extreme importance, as one function could change the parameters from other functions and disrupt the whole configuration of the network [151].

ML algorithms can be applied individually at each SON function in order to learn which parameters each function changes and by how much. Then, by integrating these different models, coordination can be achieved.

4) *Green Networks*: Another key concept of future cellular networks is energy efficiency. Current networks are dimensioned for the worst case scenario, which leads to huge amounts of power being wasted. In the future, it is expected that cellular networks can dynamically adjust their power based on its current needs. In this sense, ML algorithms can be applied, for example, in order to learn traffic patterns of individual cells and determine when is the best moment to switch the BSs on or off depending on current traffic conditions.

Another ML application can be in terms of estimating a user next position via mobility management. By learning individual user patterns and being able to predict their movement to next cells, ML algorithms can help the network to reserve resources in advance and minimize signaling between different BSs, reducing the energy consumption of the network as a whole.

C. Self-Healing

Self-healing performs a critical role in SON cellular systems, as it is responsible for detecting and mitigating the impacts that faults can have in a network. However, despite being one of the most researched areas, together with self-optimization, there is still room for improvement.

One hot topic in this area is the change of paradigms in self-healing from reactive to proactive. In order for self-healing solutions to become proactive it is essential the deployment of intelligent solutions that are able to analyze historic data and predict the behavior of the network in the future, hence, ML can play a huge role in self-healing.

By creating models from past and normal network scenarios, ML solutions can learn what are the regular network behavior and the parameters of each BS. Based on current and previous data, then predictions can be made of when and where a fault is most likely to occur in the network.

D. Data Analysis

Another key enabler in SON in future cellular networks is the concept of data analysis. As previously shown by Blondel *et al.* [266] the study of mobile phone data sets is an on-going trend and enabler of several new applications, such as social networks, determining network usage of different

areas of a country, predicting the mobility of different users, and even more robust ones, such as, urban sensing, traffic jam prevention or even detecting health and stress levels of individual users.

1) *Dark Data*: In the context of the usage of data gathered by operators, many papers have shown that despite the fact that most operators collect huge amounts of data from their subscribers on a daily basis, most of the data is still not used. In order to leverage the full-potential of SON solutions in the future, it is clear that more data need to be utilized (not collected). This data collected by operators, also known as *dark data*, can then be leveraged in order to create more robust network models and fully enable intelligence in cellular systems [8], [9].

2) *Model Identification*: Also regarding the usage of data, lies the topic of model identification. It is of extreme importance in the future to explore models inside the gathered data in order to determine patterns and explore them in order to fully configure, optimize and heal future networks. It is known that human behavior is not random, as shown by [266], and patterns such as in mobility or in traffic demand per day can already be identified in user's data. Hence, ML algorithms together with data analysis techniques can be deployed in order to learn both users and network behaviors in order to provide better QoS and QoE while also minimizing costs.

3) *Concept Drift*: Another important topic that needs to be considered for SON solutions is the idea of *concept drift*, or in other words, to consider the changes that occur in network behavior. Most of the papers presented in this survey consider one set of data for their ML algorithm and assume that the data is static. However, as it is already known, there are several patterns that can be observed in the network and in order to fully enable SON in the future, these changes that occur in the input data set must be considered in order not to misclassify or misinterpret certain situations.

For example, it is known that network traffic levels are much higher during the day than during night time. Hence, algorithms that can cope with these changes in the data, such as ML algorithms, can be deployed in order to create robust models of the network.

E. Machine Learning in 5G

5G is expected to enable whole new services and applications. In the following topics some applications of ML in 5G are discussed and a brief analysis on how ML can be applied to solve these issues is presented.

1) *Separation Between Control and Data Planes*: As future networks are expected to be more and more complex and dense, an on-going trend in current research is towards separating the control and data planes of the network. Despite this fact bringing several benefits, an additional complexity is also introduced in the system. Despite this separation, ML solutions can still be applied in both planes independently and even more robust models can be created. One example is the application of ML algorithms in the control plane, and, by learning from the control signals of the network, decisions such as CAC, mobility management or load balancing can be achieved.

From the perspective of the data plane, by learning only from the data requested by users, more robust models in terms of backhaul management, caching and resource optimization can also be achieved. Another possible application of ML regarding the split of both control and data planes is the development of ML solutions to achieve self-healing in both planes. By learning independently from data of both planes, a more general overview of the network can be achieved.

2) *Cloud Computing and Cloud RAN*: Another key enabler of future networks is the concept of cloud computing. Since some ML algorithms require lots of data and are extremely complex, one possible solution can be to use cloud computing in order to enable on-demand resources, such as computing power or even data stored in remote servers, whenever an algorithm needs such.

Other possible enabler of 5G, specially of centralized solutions, is the concept of cloud RAN, in which some processing functions of BSs can be done in a centralized way by a local controller. One possible realm in which ML algorithms can be applied is directly in these controllers. Since these controllers will process information from different BSs, models can be created in a much more optimal way, by deploying them directly to these controllers instead of each BS, and an improved performance, together with better coordination and cost reduction can be achieved.

3) *Network Function Virtualization*: Another hot topic in future networks is the concept of network function virtualization, in which its main goal is to decouple network functions from their specific hardware components, enabling a much more flexible network. By decoupling functions from their hardware, ML models can directly learn network parameters independently from hardware and provide much more generic and robust solutions.

4) *Physical Layer Management*: Another topic that is being discussed in order to enable 5G is the application of different waveforms for different application at the physical layer level. Depending on the user and applications requirements, as well as channel conditions, the network could automatically choose which are the best parameters, such as modulation and coding to transmit at that specific time slot.

In this regard, ML algorithms can be used in order to learn network and user behavior, as well as more generic aspects of the wireless channel, such as shadowing, and generate models that auto-select which waveforms and coding schemes are better for a particular application and environment.

5) *Automatic RAT Selection*: In the future, it is also expected that 5G will co-exist with different technologies in a multiple Radio Access Technology (multi-RAT) environment. Since each technology has different capabilities and provide different QoS and QoE to its end-users, one can imagine the application of ML algorithms in order to match users with different needs and requirements to the most suitable RAT. In this sense, ML solutions can learn individual user behaviors and their requirements and determine to which RAT should a user be allocated to.

6) *End-to-End Connectivity*: Current networks analyze mobile connections in terms of RAN, in which a mobile user

decides which cell to connect to based on connectivity parameters from the BS. In the future, however, as networks will become much more complex and will have to deal with several different applications at a time, this RAN vision of the network might not be sufficient and end-to-end solutions will have to be provided. One possible solution relies on the analysis of the whole mobile connection, in which not only the RAN is considered in order to determine a cell selection, but also the backhaul, so that other requirements can be considered, such as latency and capacity, for example.

In order to enable this change in paradigm, ML algorithms can be applied in order to match different users with different needs to backhaul connections that better suit them, instead of just analyzing RAN parameters.

7) *Hybrid Architecture*: Another concept that SON can also enable is the hybrid ad-hoc and cellular architecture. In current cellular systems, everything is done in a centralized way, while in ad-hoc solutions, decentralized approaches are more common. In the future, several concepts like M2M and D2D communication might transform current cellular networks in hybrid networks, requiring hybrid approaches in order to solve their issues.

One possible future research area is the application of TL in order to optimize the parameters of a hybrid network. By modeling current, centralized networks, ML algorithms can learn and build their models and then, in the future, these models can be transferred to hybrid networks, saving the operators both time and money.

8) *Learning From Machines*: Speaking of D2D and M2M communication, another aspect that can be investigated in the future is the concept of learning from machine behavior. As already stated, it is known that humans have their own patterns and are fairly predictable, but how about the machines? With the advents of IoT and the requirements of each application, it might be easier in order for ML algorithms to learn patterns and to model the communication behavior of machines, bringing bigger gains to the whole network.

Consider the case of a network of remote sensors that send collected data every week. It might be easier for ML algorithms to learn from this domain and achieve high accuracy in its predictions. With that in mind, the concept of TL can also be applied here, in which patterns learned either from humans or from other machines with different applications, can be used in order to model other machine behaviors.

F. Other Machine Learning Solutions

1) *Further Exploration of Machine Learning Algorithms*: As it could be seen from Tables II, III, IV, there are still lots of ML algorithms that have not been applied to certain SON functions. Although not every algorithm is recommended to be applied to every *self-x* function, as seen from Table VI, further exploration of ML solutions still need to be done in order to investigate their performance and determine if these methods can really work or not.

2) *Deep Learning*: One area that has seen a lot of growth in recent years and is very promising is the realm of *deep learning*, in which algorithms are fed with raw data and are

able to learn by themselves the representations needed in order to perform detection or classification [267]. Deep learning has already proved themselves to be really powerful algorithms which were able to improve state-of-the-art solutions in speech recognition, object detection and genomics, for example [267].

VIII. CONCLUSION

A survey of current ML techniques applied to SON in cellular systems was provided. In addition, not only the most popular ML techniques found in SON applications were presented and explained, but also examples from the context of cellular networks for some algorithms were given.

On top of that, this work also focused on the learning perspective of the ML algorithms and their solutions. Thus, by classifying the reviewed literature in terms of both its ML application as well as its SON function, the authors managed to develop a foundation that enables other researchers to understand the basics of the most popular ML algorithms and how they are applied in the realm of SON. Furthermore, the authors also believe that this work also enables future researchers to identify possible open issues and areas that are, currently, not well explored in terms of SON functions. The authors also present and discuss some suggestions for future research areas and outline some solutions that can be used in the future in order to enable certain SON functions.

There is a trend now in order to enable a fully autonomous and intelligent future, not only in the realm of cellular systems. The advents of smart vehicles, smart personal assistants in mobile phones, smart search algorithms, smart recommendations, all of this will require a shift and change in paradigms in future applications, and with cellular networks it is not different. In order for future networks to keep updated and on par with state-of-the-art intelligent systems a change in paradigm needs to be developed and this will most likely require the use of intelligent solutions, mainly ML algorithms.

Future networks will also require a change in the way the network is perceived. In the future, thousands of parameters will need to be configured, thousands of cells will need to be monitored and optimized at the same time and a huge amount of data will be collected, not only from humans, but also from machines. Since it is impossible for humans to deal with this amount of tasks and data, ML solutions will need to be applied in order to learn models in a relative short amount of time and to enable an autonomous and intelligent network.

REFERENCES

- [1] "5G: A technology vision," Huawei Technol. Corporat., Shenzhen, China, White Paper, 2013.
- [2] "5G vision," Samsung Electron. Corporat., Suwon, South Korea, White Paper, 2015.
- [3] "Looking ahead to 5G," Nokia Netw., Espoo, Finland, White Paper, 2014.
- [4] G. P. Fettweis, "A 5G wireless communications vision," *Microw. J.*, vol. 55, no. 12, pp. 24–36, 2012.
- [5] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.
- [6] P. Wainio and K. Seppänen, "Self-optimizing last-mile backhaul network for 5G small cells," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 232–239.

- [7] T. Alsedairy, Y. Qi, A. Imran, M. A. Imran, and B. Evans, "Self organising cloud cells: A resource efficient network densification strategy," *Trans. Emerg. Telecommun. Technol.*, vol. 26, no. 8, pp. 1096–1107, 2015. [Online]. Available: <http://dx.doi.org/10.1002/ett.2824>
- [8] N. Baldo, L. Giupponi, and J. Mangués-Bafalluy, "Big data empowered self organized networks," in *Proc. 20th Eur. Wireless Conf. Eur. Wireless*, Barcelona, Spain, May 2014, pp. 1–8.
- [9] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: How to empower SON with big data for enabling 5G," *IEEE Netw.*, vol. 28, no. 6, pp. 27–33, Nov./Dec. 2014.
- [10] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 336–361, 1st Quart., 2013.
- [11] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunications Management; Self-Organizing Networks (SON); Self-Healing Concepts and Requirements (Release 11), 3GPP TS Standard 32.541, 2012.
- [12] "3rd generation partnership project; LTE; evolved universal terrestrial radio access network (E-UTRAN); self-configuring and self-optimizing network (SON) use cases and solutions (Release 9) version 9.2.0," 3GPP, Sophia Antipolis, France, Tech. Rep. 36.902, Sep. 2010.
- [13] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication Management; Self-Organizing Networks (SON); Concepts and Requirements (Release 10), Version 10.0.0, 3GPP Standard 32.500, Jun. 2010.
- [14] S. Bi, R. Zhang, Z. Ding, and S. Cui, "Wireless communications in the era of big data," *IEEE Commun. Mag.*, vol. 53, no. 10, pp. 190–199, Oct. 2015.
- [15] M. M. S. Marwani *et al.*, "Challenges and practical implementation of self-organizing networks in LTE/LTE-advanced systems," in *Proc. Int. Conf. Inf. Technol. Multimedia (ICIM)*, Kuala Lumpur, Malaysia, 2011, pp. 1–5.
- [16] Q. Han, S. Liang, and H. Zhang, "Mobile cloud sensing, big data, and 5G networks make an intelligent and smart world," *IEEE Netw.*, vol. 29, no. 2, pp. 40–45, Mar./Apr. 2015.
- [17] K. Zheng *et al.*, "Big data-driven optimization for mobile networks toward 5G," *IEEE Netw.*, vol. 30, no. 1, pp. 44–51, Jan. 2016.
- [18] M. Musolesi, "Big mobile data mining: Good or evil?" *IEEE Internet Comput.*, vol. 18, no. 1, pp. 78–81, Jan./Feb. 2014.
- [19] M. A. Imran, A. Imran, and O. Onireti, "Participatory sensing as an enabler for self-organisation in future cellular networks," in *Proc. IOP Conf. Mater. Sci. Eng.*, vol. 51, 2013, Art. no. 012003. [Online]. Available: <http://stacks.iop.org/1757-899X/51/i=1/a=012003>
- [20] K. L. Mills, "A brief survey of self-organization in wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 7, no. 7, pp. 823–834, 2007. [Online]. Available: <http://dx.doi.org/10.1002/wcm.499>
- [21] C. Sengul, A. C. Viana, and A. Ziviani, "A survey of adaptive services to cope with dynamics in wireless self-organizing networks," *ACM Comput. Surveys*, vol. 44, no. 4, pp. 1–35, Aug. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2333112.2333118>
- [22] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1136–1159, 3rd Quart., 2013.
- [23] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1996–2018, 4th Quart., 2014.
- [24] P. Simon, *Too Big to Ignore: The Business Case for Big Data*, vol. 72. Hoboken, NJ, USA: Wiley, 2013.
- [25] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," in *Proc. Conf. Emerg. Artif. Intell. Appl. Comput. Eng. Real Word AI Syst. Appl. eHealth HCI Inf. Retrieval Pervasive Technol.*, 2007, pp. 3–24.
- [26] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning* (Springer Series in Statistics), vol. 1. New York, NY, USA: Springer-Verlag, 2001.
- [27] A. Quintero and O. Garcia, "A profile-based strategy for managing user mobility in third-generation mobile systems," *IEEE Commun. Mag.*, vol. 42, no. 9, pp. 134–139, Sep. 2004.
- [28] K. Majumdar and N. Das, "Mobile user tracking using a hybrid neural network," *Wireless Netw.*, vol. 11, no. 3, pp. 275–284, 2005. [Online]. Available: <http://dx.doi.org/10.1007/s11276-005-6611-x>
- [29] C. Yu *et al.*, "Modeling user activity patterns for next-place prediction," *IEEE Syst. J.*, vol. 11, no. 2, pp. 1060–1071, Jun. 2017.
- [30] X. Chen, F. Mériaux, and S. Valentin, "Predicting a user's next cell with supervised learning based on channel states," in *Proc. IEEE 14th Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Darmstadt, Germany, Jun. 2013, pp. 36–40.
- [31] P. Sandhir and K. Mitchell, "A neural network demand prediction scheme for resource allocation in cellular wireless systems," in *Proc. IEEE Reg. 5 Conf.*, Kansas City, MO, USA, Apr. 2008, pp. 1–6.
- [32] P. Fazio, F. D. Rango, and I. Selvaggi, "A novel passive bandwidth reservation algorithm based on neural networks path prediction in wireless environments," in *Proc. Int. Symp. Perform. Eval. Comput. Telecommun. Syst. (SPECTS)*, Ottawa, ON, Canada, Jul. 2010, pp. 38–43.
- [33] A. Adeel, H. Larijani, A. Javed, and A. Ahmadinia, "Critical analysis of learning algorithms in random neural network based cognitive engine for LTE systems," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, Glasgow, U.K., 2015, pp. 1–5.
- [34] C. A. S. Franco and J. R. B. de Marca, "Load balancing in self-organized heterogeneous LTE networks: A statistical learning approach," in *Proc. 7th IEEE Latin-Amer. Conf. Commun. (LATINCOM)*, Arequipa, Peru, 2015, pp. 1–5.
- [35] R. Narasimhan and D. C. Cox, "A handoff algorithm for wireless systems using pattern recognition," in *Proc. 9th IEEE Int. Symp. Pers. Indoor Mobile Radio Commun.*, vol. 1. Boston, MA, USA, Sep. 1998, pp. 335–339.
- [36] P. P. Bhattacharya, A. Sarkar, I. Sarkar, and S. Chatterjee, "An ANN based call handoff management scheme for mobile cellular network," *CoRR*, vol. 5, no. 6, pp. 125–135, Dec. 2013. [Online]. Available: <http://arxiv.org/abs/1401.2230>
- [37] R. Barco, V. Wille, and L. Díez, "System for automated diagnosis in cellular networks based on performance indicators," *Eur. Trans. Telecommun.*, vol. 16, no. 5, pp. 399–409, 2005.
- [38] R. M. Khanafer *et al.*, "Automated diagnosis for UMTS networks using Bayesian network approach," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2451–2461, Jul. 2008.
- [39] J. Turkka, F. Chernogorov, K. Brigatti, T. Ristaniemi, and J. Lempiäinen, "An approach for network outage detection from drive-testing databases," *J. Comput. Netw. Commun.*, vol. 2012, pp. 1–13, 2012.
- [40] C. M. Mueller, M. Kaschub, C. Blankenhorn, and S. Wanke, "A cell outage detection algorithm using neighbor cell list reports," in *Proc. Int. Workshop Self Org. Syst.*, Vienna, Austria, 2008, pp. 218–229.
- [41] W. Feng, Y. Teng, Y. Man, and M. Song, "Cell outage detection based on improved BP neural network in LTE system," in *Proc. 11th Int. Conf. Wireless Commun. Netw. Mobile Comput. (WiCOM)*, Shanghai, China, Sep. 2015, pp. 1–5.
- [42] S. Chernov, F. Chernogorov, D. Petrov, and T. Ristaniemi, "Data mining framework for random access failure detection in LTE networks," in *Proc. IEEE 25th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Washington, DC, USA, 2014, pp. 1321–1326.
- [43] V. N. Vapnik, *Statistical Learning Theory*, vol. 1. New York, NY, USA: Wiley, 1998.
- [44] S. Akoush and A. Sameh, "The use of Bayesian learning of neural networks for mobile user position prediction," in *Proc. 7th Int. Conf. Intell. Syst. Design Appl. (ISDA)*, Oct. 2007, pp. 441–446.
- [45] A. Coluccia, F. Ricciato, and P. Romirer-Maierhofer, "Bayesian estimation of network-wide mean failure probability in 3G cellular networks," in *Performance Evaluation of Computer and Communication Systems. Milestones and Future Challenges*. Berlin, Germany: Springer, 2011, pp. 167–178.
- [46] K. P. Murphy, *Naive Bayes Classifiers*, Univ. British Columbia, Vancouver, BC, Canada, 2006.
- [47] I. Rish, "An empirical study of the naive Bayes classifier," in *Proc. IJCAI Workshop Empirical Methods Artif. Intell.*, vol. 3. New York, NY, USA, 2001, pp. 41–46.
- [48] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.
- [49] O. Onireti *et al.*, "A cell outage management framework for dense heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2097–2113, Apr. 2016.
- [50] A. Zoha, A. Saeed, A. Imran, M. A. Imran, and A. Abu-Dayya, "A SON solution for sleeping cell detection using low-dimensional embedding of MDT measurements," in *Proc. IEEE 25th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Washington, DC, USA, Sep. 2014, pp. 1626–1630.

- [51] W. Xue, M. Peng, Y. Ma, and H. Zhang, "Classification-based approach for cell outage detection in self-healing heterogeneous networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Istanbul, Turkey, Apr. 2014, pp. 2822–2826.
- [52] S. Chernov, M. Cochez, and T. Ristaniemi, "Anomaly detection algorithms for the sleeping cell detection in LTE networks," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, Glasgow, U.K., May 2015, pp. 1–5.
- [53] S. Haykin, *Neural Networks—A Comprehensive Foundation*, vol. 2. Upper Saddle River, NJ, USA: Prentice-Hall, 2004.
- [54] E. Gelenbe, "Random neural networks with negative and positive signals and product form solution," *Neural Comput.*, vol. 1, no. 4, pp. 502–510, 1989.
- [55] E. Ostlin, H.-J. Zepernick, and H. Suzuki, "Macrocell radio wave propagation prediction using an artificial neural network," in *Proc. IEEE 60th Veh. Technol. Conf. (VTC Fall)*, vol. 1. Los Angeles, CA, USA, Sep. 2004, pp. 57–61.
- [56] Y. Zang, F. Ni, Z. Feng, S. Cui, and Z. Ding, "Wavelet transform processing for cellular traffic prediction in machine learning networks," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process. (ChinaSIP)*, Chengdu, China, Jul. 2015, pp. 458–462.
- [57] I. Railean, C. Stolojescu, S. Moga, and P. Lenca, "WIMAX traffic forecasting based on neural networks in wavelet domain," in *Proc. 4th Int. Conf. Res. Challenges Inf. Sci. (RCIS)*, Nice, France, 2010, pp. 443–452.
- [58] T. Edwards, D. Tansley, R. Frank, and N. Davey, "Traffic trends analysis using neural networks," in *Proc. Int. Workshop Appl. Neural Netw. Telecommun.*, 1997, pp. 157–164.
- [59] A. Quintero, "A user pattern learning strategy for managing users' mobility in UMTS networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 6, pp. 552–566, Nov./Dec. 2005.
- [60] S. Premchaisawatt and N. Ruangchaijaturon, "Enhancing indoor positioning based on partitioning cascade machine learning models," in *Proc. 11th Int. Conf. Elect. Eng./Electron. Comput. Telecommun. Inf. Technol. (ECTI-CON)*, Nakhon Ratchasima, Thailand, May 2014, pp. 1–5.
- [61] J. Capka and R. Boutaba, "Mobility prediction in wireless networks using neural networks," in *Proc. IFIP/IEEE Int. Conf. Manag. Multimedia Netw. Services*, San Diego, CA, USA, 2004, pp. 320–333.
- [62] M. Stoyanova and P. Mahonen, "A next-move prediction algorithm for implementation of selective reservation concept in wireless networks," in *Proc. IEEE 18th Int. Symp. Pers. Indoor Mobile Radio Commun.*, Athens, Greece, Sep. 2007, pp. 1–5.
- [63] M. Vukovic, I. Lovrek, and D. Jevtic, "Predicting user movement for advanced location-aware services," in *Proc. 15th Int. Conf. Softw. Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2007, pp. 1–5.
- [64] M. Ekpenyong, J. Isabona, and E. Isong, "Handoffs decision optimization of mobile cellular networks," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Las Vegas, NV, USA, Dec. 2015, pp. 697–702.
- [65] Z. Ali, N. Baldo, J. Mangués-Bafalluy, and L. Giupponi, "Machine learning based handover management for improved QoE in LTE," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS)*, Istanbul, Turkey, Apr. 2016, pp. 794–798.
- [66] R. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, vol. ASSPM-4, no. 2, pp. 4–22, Apr. 1987.
- [67] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory (COLT)*, Pittsburgh, PA, USA, 1992, pp. 144–152. [Online]. Available: <http://doi.acm.org/10.1145/130385.130401>
- [68] V.-S. Feng and S. Y. Chang, "Determination of wireless networks parameters through parallel hierarchical support vector machines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 3, pp. 505–512, Mar. 2012.
- [69] G. F. Ciocarlie, U. Lindqvist, S. Novaczki, and H. Sanneck, "Detecting anomalies in cellular networks using an ensemble method," in *Proc. 9th Int. Conf. Netw. Service Manag. (CNSM)*, Zürich, Switzerland, Oct. 2013, pp. 171–174.
- [70] A. Zoha, A. Saeed, A. Imran, M. A. Imran, and A. Abu-Dayya, "Data-driven analytics for automated cell outage detection in self-organizing networks," in *Proc. 11th Int. Conf. Design Rel. Commun. Netw. (DRCN)*, Kansas City, MO, USA, Mar. 2015, pp. 203–210.
- [71] A. Zoha, A. Saeed, A. Imran, M. A. Imran, and A. Abu-Dayya, "A learning-based approach for autonomous outage detection and coverage optimization," *Trans. Emerg. Telecommun. Technol.*, vol. 27, no. 3, pp. 439–450, 2016. [Online]. Available: <http://dx.doi.org/10.1002/ett.2971>
- [72] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.
- [73] H. Y. Lateef, A. Imran, and A. Abu-dayya, "A framework for classification of self-organising network conflicts and coordination algorithms," in *Proc. IEEE 24th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, London, U.K., Sep. 2013, pp. 2898–2903.
- [74] J. Puttonen, J. Turkka, O. Alanen, and J. Kurjenniemi, "Coverage optimization for minimization of drive tests in LTE with extended RLF reporting," in *Proc. 21st Annu. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun.*, Istanbul, Turkey, Sep. 2010, pp. 1764–1768.
- [75] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, 1992.
- [76] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–58, Mar. 1997. [Online]. Available: <http://doi.acm.org/10.1145/245108.245121>
- [77] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, pp. 1–35. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-85820-3_1
- [78] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. 14th Conf. Uncertainty Artif. Intell.*, Madison, WI, USA, 1998, pp. 43–52. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2074094.2074100>
- [79] W. Wang, J. Zhang, and Q. Zhang, "Cooperative cell outage detection in self-organizing femtocell networks," in *Proc. IEEE INFOCOM*, Turin, Italy, 2013, pp. 782–790.
- [80] W. Wang and Q. Zhang, "Local cooperation architecture for self-healing femtocell networks," *IEEE Wireless Commun.*, vol. 21, no. 2, pp. 42–49, Apr. 2014.
- [81] W. Wang, Q. Liao, and Q. Zhang, "COD: A cooperative cell outage detection architecture for self-organizing femtocell networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 6007–6014, Nov. 2014.
- [82] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.
- [83] T. Binzer and F. M. Landstorfer, "Radio network planning with neural networks," in *Proc. 52nd Veh. Technol. Conf. (IEEE-VTS Fall VTC)*, vol. 2. Boston, MA, USA, 2000, pp. 811–817.
- [84] M. Peng, D. Liang, Y. Wei, J. Li, and H.-H. Chen, "Self-configuration and self-optimization in LTE-advanced heterogeneous networks," *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 36–45, May 2013.
- [85] K. Hamidouche, W. Saad, and M. Debbah, "Many-to-many matching games for proactive social-caching in wireless small cell networks," in *Proc. 12th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, 2014, pp. 569–574.
- [86] P. Blasco and D. Gündüz, "Learning-based optimization of cache content in a small cell base station," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, 2014, pp. 1897–1903.
- [87] D. Kumar, N. Kanagaraj, and R. Srilakshmi, "Harmonized Q-learning for radio resource management in LTE based networks," in *Proc. ITU Kaleidoscope Build. Sustain. Communities (K)*, Kyoto, Japan, 2013, pp. 1–8.
- [88] P. Savazzi and L. Favalli, "Dynamic cell sectorization using clustering algorithms," in *Proc. IEEE 65th Veh. Technol. Conf. (VTC Spring)*, 2007, pp. 604–608.
- [89] N. Sinclair, D. Harle, I. A. Glover, J. Irvine, and R. C. Atkinson, "An advanced SOM algorithm applied to handover management within LTE," *IEEE Trans. Veh. Technol.*, vol. 62, no. 5, pp. 1883–1894, Jun. 2013.
- [90] M. Stoyanova and P. Mahonen, "Algorithmic approaches for vertical handoff in heterogeneous wireless environment," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Hong Kong, Mar. 2007, pp. 3780–3785.
- [91] A. Chakraborty, L. E. Ortiz, and S. R. Das, "Network-side positioning of cellular-band devices with minimal effort," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Hong Kong, Apr. 2015, pp. 2767–2775.
- [92] S. Basso, M. Jaber, M. A. Imran, and P. Xiao, "Load aware self-organising user-centric dynamic CoMP clustering for 5G networks," *IEEE Access*, vol. 4, pp. 2895–2906, 2016.
- [93] K. Raivio, O. Simula, J. Laiho, and P. Lehtimäki, "Analysis of mobile radio access network using the self-organizing map," in *Proc. IFIP/IEEE 8th Int. Symp. Integr. Netw. Manag.*, Colorado Springs, CO, USA, Mar. 2003, pp. 439–451.
- [94] K. Raivio, O. Simula, and J. Laiho, "Neural analysis of mobile radio access network," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, San Jose, CA, USA, 2001, pp. 457–464.

- [95] M. Kylvaja *et al.*, "Trial report on self-organizing map based analysis tool for radio networks [GSM applications]," in *Proc. IEEE 59th Veh. Technol. Conf. (VTC Spring)*, vol. 4. Milan, Italy, May 2004, pp. 2365–2369.
- [96] P. Sukhawatchani and W. Usaha, "Performance evaluation of anomaly detection in cellular core networks using self-organizing map," in *Proc. 5th Int. Conf. Elect. Eng./Electron. Comput. Telecommun. Inf. Technol. (ECTI-CON)*, vol. 1. 2008, pp. 361–364.
- [97] P. Fiadino, A. Dalconzo, M. Schiavone, and P. Casas, "RCATool—A framework for detecting and diagnosing anomalies in cellular networks," in *Proc. 27th Int. Teletraffic Congr. (ITC)*, Ghent, Belgium, 2015, pp. 194–202.
- [98] P. Szilágyi and S. Nováczki, "An automatic detection and diagnosis framework for mobile communication systems," *IEEE Trans. Netw. Service Manag.*, vol. 9, no. 2, pp. 184–197, Jun. 2012.
- [99] S. Nováczki, "An improved anomaly detection and diagnosis framework for mobile network operators," in *Proc. 9th Int. Conf. Design Rel. Commun. Netw. (DRCN)*, Budapest, Hungary, 2013, pp. 234–241.
- [100] A. D'Alconzo, A. Coluccia, F. Ricciato, and P. Romirer-Maierhofer, "A distribution-based approach to anomaly detection and application to 3G mobile traffic," in *Proc. GLOBECOM*, Honolulu, HI, USA, 2009, pp. 1–8.
- [101] I.-H. Bae and S. Olariu, "A weighted-dissimilarity-based anomaly detection method for mobile wireless networks," in *Proc. Int. Conf. Comput. Sci. Eng. (CSE)*, vol. 1. Vancouver, BC, Canada, Aug. 2009, pp. 29–34.
- [102] A. Bouillard, A. Junier, and B. Ronot, "Hidden anomaly detection in telecommunication networks," in *Proc. 8th Int. Conf. Netw. Service Manag.*, Las Vegas, NV, USA, 2012, pp. 82–90.
- [103] S. Chernov, D. Petrov, and T. Ristaniemi, "Location accuracy impact on cell outage detection in LTE-A networks," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Dubrovnik, Croatia, Aug. 2015, pp. 1162–1167.
- [104] I. de-la Bandera, R. Barco, P. Munoz, and I. Serrano, "Cell outage detection based on handover statistics," *IEEE Commun. Lett.*, vol. 19, no. 7, pp. 1189–1192, Jul. 2015.
- [105] P. Muñoz, R. Barco, I. Serrano, and A. Gómez-Andrades, "Correlation-based time-series analysis for cell degradation detection in SON," *IEEE Commun. Lett.*, vol. 20, no. 2, pp. 396–399, Feb. 2016.
- [106] Y. Ma, M. Peng, W. Xue, and X. Ji, "A dynamic affinity propagation clustering algorithm for cell outage detection in self-healing networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Shanghai, China, Apr. 2013, pp. 2266–2270.
- [107] F. Chernogorov, T. Ristaniemi, K. Brigatti, and S. Chernov, "N-gram analysis for sleeping cell detection in LTE networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 4439–4443.
- [108] U. S. Hashmi, A. Darbandi, and A. Imran, "Enabling proactive self-healing by data mining network failure logs," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Santa Clara, CA, USA, Jan. 2017, pp. 511–517.
- [109] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, no. 1, pp. 1–6, 1998.
- [110] C. J. Debono and J. K. Buhagiar, "Cellular network coverage optimization through the application of self-organizing neural networks," in *Proc. IEEE 62nd Veh. Technol. Conf. (VTC Fall)*, vol. 4. Dallas, TX, USA, 2005, pp. 2158–2162.
- [111] A. Gómez-Andrades, R. Barco, P. Muñoz, and I. Serrano, "Data analytics for diagnosing the RF condition in self-organizing networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 6, pp. 1587–1600, Jun. 2017.
- [112] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surveys*, vol. 41, no. 3, p. 15, 2009.
- [113] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, 2004.
- [114] M. Thottan, G. Liu, and C. Ji, *Anomaly Detection Approaches for Communication Networks*. London, U.K.: Springer, 2010, pp. 239–261. [Online]. Available: http://dx.doi.org/10.1007/978-1-84882-765-3_11
- [115] Q. Liao, M. Wiczowski, and S. Stańczak, "Toward cell outage detection with composite hypothesis testing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, ON, Canada, 2012, pp. 4883–4887.
- [116] I. de-la Bandera, P. Munoz, I. Serrano, and R. Barco, "Improving cell outage management through data analysis," *IEEE Wireless Commun.*, to be published, doi: 10.1109/MWC.2017.1600076WC.
- [117] K. Ogata and Y. Yang, *Modern Control Engineering*, 4th ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1970.
- [118] J. Li and R. Jantti, "On the study of self-configuration neighbour cell list for mobile WiMAX," in *Proc. Int. Conf. Next Gener. Mobile Appl. Services Technol. (NGMAST)*, Cardiff, U.K., 2007, pp. 199–204.
- [119] F. Parodi, M. Kylvaja, G. Alford, J. Li, and J. Pradas, "An automatic procedure for neighbor cell list definition in cellular networks," in *Proc. IEEE Int. Symp. World Wireless Mobile Multimedia Netw.*, Jun. 2007, pp. 1–6.
- [120] D. Soldani and I. Ore, "Self-optimizing neighbor cell list for UTRA FDD networks using detected set reporting," in *Proc. IEEE 65th Veh. Technol. Conf. (VTC Spring)*, Espoo, Finland, 2007, pp. 694–698.
- [121] S. S. Mwanje, N. Zia, and A. Mitschele-Thiel, "Self-organized handover parameter configuration for LTE," in *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, Paris, France, Aug. 2012, pp. 26–30.
- [122] H. Claussen, L. T. W. Ho, and L. G. Samuel, "Self-optimization of coverage for femtocell deployments," in *Proc. Wireless Telecommun. Symp. (WTS)*, Pomona, CA, USA, Apr. 2008, pp. 278–285.
- [123] X. Zhao and P. Chen, "Improving UE SINR and networks energy efficiency based on femtocell self-optimization capability," in *Proc. Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Istanbul, Turkey, 2014, pp. 155–160.
- [124] R. Joyce and L. Zhang, "Self organising network techniques to maximise traffic offload onto a 3G/WCDMA small cell network using MDT UE measurement reports," in *Proc. IEEE Glob. Commun. Conf.*, Austin, TX, USA, Dec. 2014, pp. 2212–2217.
- [125] A. Gerdenitsch, S. Jakl, Y. Y. Chong, and M. Toeltsch, "A rule-based algorithm for common pilot channel and antenna tilt optimization in UMTS FDD networks," *ETRI J.*, vol. 26, no. 5, pp. 437–442, 2004.
- [126] D. Fagen, P. A. Vicharelli, and J. Weitzen, "Automated wireless coverage optimization with controlled overlap," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2395–2403, Jul. 2008.
- [127] A. Engels *et al.*, "Autonomous self-optimization of coverage and capacity in LTE cellular networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 5, pp. 1989–2004, Jun. 2013.
- [128] J.-H. Yun and K. G. Shin, "CTRL: A self-organizing femtocell management architecture for co-channel deployment," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, Chicago, IL, USA, 2010, pp. 61–72.
- [129] I. Karla, "Distributed algorithm for self organizing LTE interference coordination," in *Proc. Int. Conf. Mobile Netw. Manag.*, Athens, Greece, 2009, pp. 119–128.
- [130] M. Mehta, N. Rane, A. Karandikar, M. A. Imran, and B. G. Evans, "A self-organized resource allocation scheme for heterogeneous macro-femto networks," *Wireless Commun. Mobile Comput.*, vol. 16, no. 3, pp. 330–342, 2016. [Online]. Available: <http://dx.doi.org/10.1002/wcm.2518>
- [131] I. Joe and S. Hong, "A mobility-based prediction algorithm for vertical handover in hybrid wireless networks," in *Proc. 2nd IEEE/IFIP Int. Workshop Broadband Conver. Netw. (BcN)*, Munich, Germany, May 2007, pp. 1–5.
- [132] G. Hui and P. Legg, "Soft metric assisted mobility robustness optimization in LTE networks," in *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, Paris, France, Aug. 2012, pp. 1–5.
- [133] A. Schröder, H. Lundqvist, and G. Nunzi, "Distributed self-optimization of handover for the long term evolution," in *Proc. Int. Workshop Self Org. Syst.*, Vienna, Austria, 2008, pp. 281–286.
- [134] D.-W. Lee, G.-T. Gil, and D.-H. Kim, "A cost-based adaptive handover hysteresis scheme to minimize the handover failure rate in 3GPP LTE system," *EURASIP J. Wireless Commun. Netw.*, vol. 2010, no. 1, p. 6, 2010.
- [135] K. Kitagawa, T. Komine, T. Yamamoto, and S. Konishi, "A handover optimization algorithm with mobility robustness for LTE systems," in *Proc. IEEE 22nd Int. Symp. Pers. Indoor Mobile Radio Commun.*, Toronto, ON, Canada, Sep. 2011, pp. 1647–1651.
- [136] L. Ewe and H. Bakker, "Base station distributed handover optimization in LTE self-organizing networks," in *Proc. IEEE 22nd Int. Symp. Pers. Indoor Mobile Radio Commun.*, Toronto, ON, Canada, Sep. 2011, pp. 243–247.
- [137] I. Balan, T. Jansen, B. Sas, I. Moerman, and T. Kürner, "Enhanced weighted performance based handover optimization in LTE," in *Proc. Future Netw. Mobile Summit (FutureNetw)*, Warsaw, Poland, Jun. 2011, pp. 1–8.
- [138] Q. Song, Z. Wen, X. Wang, L. Guo, and R. Yu, "Time-adaptive vertical handoff triggering methods for heterogeneous systems," in *Proc. Int. Workshop Adv. Parallel Process. Technol.*, 2009, pp. 302–312.
- [139] A. Awada, B. Wegmann, I. Viering, and A. Klein, "A SON-based algorithm for the optimization of inter-RAT handover parameters," *IEEE Trans. Veh. Technol.*, vol. 62, no. 5, pp. 1906–1923, Jun. 2013.

- [140] A. Beletchi, F. Huang, H. Zhuang, and J. Zha, "Mobility self-optimization in LTE networks based on adaptive control theory," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Atlanta, GA, USA, 2013, pp. 87–92.
- [141] J. Alonso-Rubio, "Self-optimization for handover oscillation control in LTE," in *Proc. IEEE Netw. Oper. Manag. Symp. (NOMS)*, Osaka, Japan, Apr. 2010, pp. 950–953.
- [142] T. Jansen, I. Balan, J. Turk, I. Moerman, and T. Kurner, "Handover parameter optimization in LTE self-organizing networks," in *Proc. IEEE 72nd Veh. Technol. Conf. Fall (VTC Fall)*, Ottawa, ON, Canada, Sep. 2010, pp. 1–5.
- [143] D. Soldani, G. Alford, F. Parodi, and M. Kylvaja, "An autonomic framework for self-optimizing next generation mobile networks," in *Proc. IEEE Int. Symp. World Wireless Mobile Multimedia Netw.*, Espoo, Finland, Jun. 2007, pp. 1–6.
- [144] C.-L. Lee, W.-S. Su, K.-A. Tang, and W.-I. Chao, "Design of handover self-optimization using big data analytics," in *Proc. 16th Asia-Pac. Netw. Oper. Manag. Symp. (APNOMS)*, Hsinchu, Taiwan, Sep. 2014, pp. 1–5.
- [145] I. Viering, M. Döttling, and A. Lobinger, "A mathematical perspective of self-optimizing wireless networks," in *Proc. IEEE Int. Conf. Commun.*, Dresden, Germany, Jun. 2009, pp. 1–6.
- [146] A. Lobinger, S. Stefanski, T. Jansen, and I. Balan, "Load balancing in downlink LTE self-optimizing networks," in *Proc. IEEE 71st Veh. Technol. Conf. (VTC Spring)*, Taipei, Taiwan, May 2010, pp. 1–5.
- [147] B. Fan, S. Leng, and K. Yang, "A dynamic bandwidth allocation algorithm in mobile networks with big data of users and networks," *IEEE Netw.*, vol. 30, no. 1, pp. 6–10, Jan./Feb. 2016.
- [148] A. Liakopoulos *et al.*, "Applying distributed monitoring techniques in autonomic networks," in *Proc. IEEE Globecom Workshops*, Miami, FL, USA, 2010, pp. 498–502.
- [149] L.-T. Lee, C.-F. Wu, D.-F. Tao, and K.-Y. Liu, "A cell-based call admission control policy with time series prediction and throttling mechanism for supporting QoS in wireless cellular networks," in *Proc. Int. Symp. Commun. Inf. Technol.*, Bangkok, Thailand, Oct. 2006, pp. 88–93.
- [150] A. Tall, R. Combes, Z. Altman, and E. Altman, "Distributed coordination of self-organizing mechanisms in communication networks," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 4, pp. 328–337, Dec. 2014.
- [151] H. Y. Lateef, A. Imran, M. A. Imran, L. Giupponi, and M. Dohler, "LTE-advanced self-organizing network conflicts and coordination algorithms," *IEEE Wireless Commun.*, vol. 22, no. 3, pp. 108–117, Jun. 2015.
- [152] I. Karla, "Resolving SON interactions via self-learning prediction in cellular wireless networks," in *Proc. 8th Int. Conf. Wireless Commun. Netw. Mobile Comput. (WiCOM)*, Shanghai, China, Sep. 2012, pp. 1–6.
- [153] N. Tcholtchev and R. Chaparadza, "Autonomic fault-management and resilience from the perspective of the network operation personnel," in *Proc. IEEE Globecom Workshops*, Miami, FL, USA, 2010, pp. 469–474.
- [154] E. H. B. Bramah and H. A. Ali, "Self healing in long term evolution (LTE) (development of a cell outage compensation algorithm)," in *Proc. Conf. Basic Sci. Eng. Stud. (SGCAC)*, Khartoum, Sudan, Feb. 2016, pp. 75–79.
- [155] M. Amirjoo *et al.*, "Cell outage management in LTE networks," in *Proc. 6th Int. Symp. Wireless Commun. Syst.*, Sep. 2009, pp. 600–604.
- [156] F.-Q. Li, X.-S. Qiu, L.-M. Meng, H. Zhang, and W. Gu, "Achieving cell outage compensation in radio access network with automatic network management," in *Proc. IEEE GLOBECOM Workshops (GC Wkshps)*, Houston, TX, USA, Dec. 2011, pp. 673–677.
- [157] L. Kayili and E. Sousa, "Cell outage compensation for irregular cellular networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Istanbul, Turkey, Apr. 2014, pp. 1850–1855.
- [158] F. Li, X. Qiu, Z. Tian, B. Wang, and L. Meng, "Adjusting electrical downtilt based mechanism of autonomous cell outage compensation," in *Proc. IET Int. Conf. Commun. Technol. Appl. (ICCTA)*, Beijing, China, Oct. 2011, pp. 389–394.
- [159] E. Chu, I. Bang, S. H. Kim, and D. K. Sung, "Self-organizing and self-healing mechanisms in cooperative small-cell networks," in *Proc. IEEE 24th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, London, U.K., Sep. 2013, pp. 1576–1581.
- [160] O. Onireti, A. Imran, M. A. Imran, and R. Tafazolli, "Cell outage detection in heterogeneous networks with separated control and data plane," in *Proc. 20th Eur. Wireless Conf. Eur. Wireless*, Barcelona, Spain, May 2014, pp. 1–6.
- [161] M. Amirjoo, L. Jorgueski, R. Litjens, and L. C. Schmelz, "Cell outage compensation in LTE networks: Algorithms and performance assessment," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, Yokohama, Japan, May 2011, pp. 1–5.
- [162] T. J. Ross, *Fuzzy Logic With Engineering Applications*. New York, NY, USA: Wiley, 2009.
- [163] J. M. Mendel, "Fuzzy logic systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, no. 3, pp. 345–377, Mar. 1995.
- [164] N. Farzaneh and M. H. Y. Moghaddam, "Virtual topology reconfiguration of WDM optical networks using fuzzy logic control," in *Proc. Int. Symp. Telecommun. (IST)*, Tehran, Iran, Aug. 2008, pp. 504–509.
- [165] S. Luna-Ramirez, M. Toril, F. Ruiz, and M. Fernandez-Navarro, "Adjustment of a fuzzy logic controller for IS-HO parameters in a heterogeneous scenario," in *Proc. 14th IEEE Mediterr. Electrotech. Conf. (MELECON)*, Ajaccio, France, May 2008, pp. 29–34.
- [166] C. Werner, J. Voigt, S. Khattak, and G. Fettweis, "Handover parameter optimization in WCDMA using fuzzy controlling," in *Proc. IEEE 18th Int. Symp. Pers. Indoor Mobile Radio Commun.*, Athens, Greece, Sep. 2007, pp. 1–5.
- [167] M. McGuire and V. K. Bhargava, "A robust fuzzy logic handoff algorithm," in *Proc. IEEE Can. Conf. Elect. Comput. Eng. Innov. Voyage Disc.*, vol. 2, St. John's, NL, Canada, May 1997, pp. 796–799.
- [168] L. Barolli, F. Xhafa, A. Duresi, A. Koyama, and M. Takizawa, "An intelligent handoff system for wireless cellular networks using fuzzy logic and random walk model," in *Proc. Int. Conf. Complex Intell. Softw. Intensive Syst. (CISIS)*, Barcelona, Spain, Mar. 2008, pp. 5–11.
- [169] A. Ezzouhairi, A. Quintero, and S. Pierre, "A fuzzy decision making strategy for vertical handoffs," in *Proc. IEEE Can. Conf. Elect. Comput. Eng. (CCECE)*, Niagara Falls, ON, Canada, 2008, pp. 000583–000588.
- [170] M. S. Dang, A. Prakash, D. K. Anvekar, D. Kapoor, and R. Shorey, "Fuzzy logic based handoff in wireless networks," in *Proc. IEEE 51st Veh. Technol. Conf. (VTC Spring)*, vol. 3, Tokyo, Japan, 2000, pp. 2375–2379.
- [171] M. Toril and V. Wille, "Optimization of handover parameters for traffic sharing in GERAN," *Wireless Pers. Commun.*, vol. 47, no. 3, pp. 315–336, 2008.
- [172] P. Muñoz, R. Barco, and I. de la Bandera, "On the potential of handover parameter optimization for self-organizing networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 5, pp. 1895–1905, Jun. 2013.
- [173] F. Bouali, K. Moessner, and M. Fitch, "A context-aware user-driven framework for network selection in 5G multi-RAT environments," in *Proc. IEEE 84th Veh. Technol. Conf. (VTC Fall)*, Montreal, QC, Canada, Sep. 2016, pp. 1–7.
- [174] J. Rodriguez, I. D. la Bandera, P. Munoz, and R. Barco, "Load balancing in a realistic urban scenario for LTE networks," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, Yokohama, Japan, May 2011, pp. 1–5.
- [175] P. Muñoz, R. Barco, J. M. Ruiz-Avilés, I. de la Bandera, and A. Aguilar, "Fuzzy rule-based reinforcement learning for load balancing techniques in enterprise LTE femtocells," *IEEE Trans. Veh. Technol.*, vol. 62, no. 5, pp. 1962–1973, Jun. 2013.
- [176] P. Kiran, M. G. Jibukumar, and C. V. Premkumar, "Resource allocation optimization in LTE-A/5G networks using big data analytics," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Kota Kinabalu, Malaysia, 2016, pp. 254–259.
- [177] J. Ye, X. Shen, and J. W. Mark, "Call admission control in wideband CDMA cellular networks by using fuzzy logic," in *Proc. IEEE Wireless Commun. Netw. (WCNC)*, vol. 3, New Orleans, LA, USA, Mar. 2003, pp. 1538–1543.
- [178] S. B. ZahirAzami, G. Yekrangian, and M. Spencer, "Load balancing and call admission control in UMTS-RNC, using fuzzy logic," in *Proc. Int. Conf. Commun. Technol. (ICCT)*, vol. 2, Beijing, China, Apr. 2003, pp. 790–793.
- [179] H. Nan, H. Zhiqiang, N. Kai, and W. Wei-Ling, "Connection admission control for OFDM cellular networks by using fuzzy logic," in *Proc. Int. Conf. Commun. Technol.*, Guilin, China, Nov. 2006, pp. 1–4.
- [180] A. K. Mukhopadhyay, S. Chatterjee, S. Saha, S. Ghose, and D. Saha, "An efficient call admission control scheme on overlay networks using fuzzy logic," in *Proc. IEEE 3rd Int. Symp. Adv. Netw. Telecommun. Syst. (ANTS)*, New Delhi, India, Dec. 2009, pp. 1–3.
- [181] S. V. Truong, L. L. Hung, and H. N. Thanh, "A fuzzy logic call admission control scheme in multi-class traffic cellular mobile networks," in *Proc. Int. Symp. Comput. Commun. Control Autom. (3CA)*, vol. 1, Tainan, Taiwan, May 2010, pp. 330–333.

- [182] T. Inaba, S. Sakamoto, T. Oda, and L. Barolli, "Performance evaluation of a secure call connection admission control for wireless cellular networks using fuzzy logic," in *Proc. 10th Int. Conf. Broadband Wireless Comput. Commun. Appl. (BWCCA)*, Krakow, Poland, Nov. 2015, pp. 437–441.
- [183] Q. Liao and S. Stanczak, "Network state awareness and proactive anomaly detection in self-organizing networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, San Diego, CA, USA, Dec. 2015, pp. 1–6.
- [184] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, vol. 1. Cambridge, MA, USA: MIT Press, 1998.
- [185] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, 1996. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1622737.1622748>
- [186] M. N. U. Islam and A. Mitschele-Thiel, "Reinforcement learning strategies for self-organized coverage and capacity optimization," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Shanghai, China, Apr. 2012, pp. 2818–2823.
- [187] R. Razavi, S. Klein, and H. Claussen, "Self-optimization of capacity and coverage in LTE networks using a fuzzy reinforcement learning approach," in *Proc. 21st Annu. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun.*, Istanbul, Turkey, Sep. 2010, pp. 1865–1870.
- [188] R. Razavi, S. Klein, and H. Claussen, "A fuzzy reinforcement learning approach for self-optimization of coverage in LTE networks," *Bell Labs Tech. J.*, vol. 15, no. 3, pp. 153–175, Dec. 2010.
- [189] M. S. ElBamby, M. Bennis, W. Saad, and M. Latva-Aho, "Content-aware user clustering and caching in wireless small cell networks," in *Proc. 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Barcelona, Spain, 2014, pp. 945–949.
- [190] M. Jaber, M. Imran, R. Tafazolli, and A. Tukmanov, "An adaptive backhaul-aware cell range extension approach," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, London, U.K., 2015, pp. 74–79.
- [191] M. Jaber, M. A. Imran, R. Tafazolli, and A. Tukmanov, "A distributed SON-based user-centric backhaul provisioning scheme," *IEEE Access*, vol. 4, pp. 2314–2330, 2016.
- [192] M. Jaber, M. A. Imran, R. Tafazolli, and A. Tukmanov, "A multiple attribute user-centric backhaul provisioning scheme using distributed SON," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1–6.
- [193] M. Jaber, M. A. Imran, R. Tafazolli, and A. Tukmanov, "Energy-efficient SON-based user-centric backhaul scheme," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.
- [194] M. Bennis and D. Niyato, "A Q-learning based approach to interference avoidance in self-organized femtocell networks," in *Proc. IEEE Globecom Workshops*, Miami, FL, USA, Dec. 2010, pp. 706–710.
- [195] M. Dirani and Z. Altman, "A cooperative reinforcement learning approach for inter-cell interference coordination in OFDMA cellular networks," in *Proc. 8th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, Avignon, France, May 2010, pp. 170–176.
- [196] S. S. Mwanje and A. Mitschele-Thiel, "Distributed cooperative Q-learning for mobility-sensitive handover optimization in LTE SON," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Funchal, Portugal, Jun. 2014, pp. 1–6.
- [197] S. S. Mwanje, L. C. Schmelz, and A. Mitschele-Thiel, "Cognitive cellular networks: A Q-learning framework for self-organizing networks," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 1, pp. 85–98, Mar. 2016.
- [198] C. Dhahri and T. Ohtsuki, "Adaptive Q-learning cell selection method for open-access femtocell networks: Multi-user case," *IEICE Trans. Commun.*, vol. 97, no. 8, pp. 1679–1688, 2014.
- [199] P. Munoz, R. Barco, I. de la Bandera, M. Toril, and S. Luna-Ramirez, "Optimization of a fuzzy logic controller for handover-based load balancing," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, Yokohama, Japan, May 2011, pp. 1–5.
- [200] S. S. Mwanje and A. Mitschele-Thiel, "A Q-learning strategy for LTE mobility load balancing," in *Proc. IEEE 24th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, London, U.K., Sep. 2013, pp. 2154–2158.
- [201] T. Kudo and T. Ohtsuki, "Q-learning based cell selection for UE outage reduction in heterogeneous networks," in *Proc. IEEE 80th Veh. Technol. Conf. (VTC Fall)*, Vancouver, BC, Canada, 2014, pp. 1–5.
- [202] M. Dirani and Z. Altman, "Self-organizing networks in next generation radio access networks: Application to fractional power control," *Comput. Netw.*, vol. 55, no. 2, pp. 431–438, 2011.
- [203] E. Alexandri, G. Martinez, and D. Zeghlache, "A distributed reinforcement learning approach to maximize resource utilization and control handover dropping in multimedia wireless networks," in *Proc. 13th IEEE Int. Symp. Pers. Indoor Mobile Radio Commun.*, vol. 5, 2002, pp. 2249–2253.
- [204] P.-Y. Kong, D. Panaitopol, and A. Dhahi, "Reinforcement learning approach to dynamic activation of base station resources in wireless networks," in *Proc. PIMRC*, London, U.K., 2013, pp. 3264–3268.
- [205] A. Galindo-Serrano, L. Giupponi, and G. Auer, "Distributed learning in multiuser OFDMA femtocell networks," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, Yokohama, Japan, 2011, pp. 1–6.
- [206] D. Liu and Y. Zhang, "A self-learning adaptive critic approach for call admission control in wireless cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 3, Anchorage, AK, USA, May 2003, pp. 1853–1857.
- [207] M. Miozzo, L. Giupponi, M. Rossi, and P. Dini, "Switch-on/off policies for energy harvesting small cells through distributed Q-learning," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.
- [208] A. Saeed, O. G. Aliu, and M. A. Imran, "Controlling self healing cellular networks using fuzzy logic," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Shanghai, China, Apr. 2012, pp. 3080–3084.
- [209] J. Moysen and L. Giupponi, "A reinforcement learning based solution for self-healing in LTE networks," in *Proc. IEEE 80th Veh. Technol. Conf. (VTC Fall)*, Vancouver, BC, Canada, Sep. 2014, pp. 1–6.
- [210] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [211] A. Mohamed *et al.*, "Mobility prediction for handover management in cellular networks with control/data separation," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 3939–3944.
- [212] H. Si, Y. Wang, J. Yuan, and X. Shan, "Mobility prediction in cellular network using hidden Markov model," in *Proc. 7th IEEE Consum. Commun. Netw. Conf.*, Las Vegas, NV, USA, Jan. 2010, pp. 1–5.
- [213] P. Fazio, M. Tropea, and S. Marano, "A distributed hand-over management and pattern prediction algorithm for wireless networks with mobile hosts," in *Proc. 9th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jul. 2013, pp. 294–298.
- [214] H. Farooq and A. Imran, "Spatiotemporal mobility prediction in proactive self-organizing cellular networks," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 370–373, Feb. 2017.
- [215] A. Mohamed, O. Onireti, M. A. Imran, A. Imran, and R. Tafazolli, "Predictive and core-network efficient RRC signalling for active state handover in RANs with control/data separation," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1423–1436, Mar. 2017.
- [216] A. F. Santamaria and A. Lupia, "A new call admission control scheme based on pattern prediction for mobile wireless cellular networks," in *Proc. Wireless Telecommun. Symp. (WTS)*, New York, NY, USA, Apr. 2015, pp. 1–6.
- [217] M. Peng and W. Wang, "An adaptive energy saving mechanism in the wireless packet access network," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Las Vegas, NV, USA, 2008, pp. 1536–1540.
- [218] H. Farooq, M. S. Parwez, and A. Imran, "Continuous time Markov chain based reliability analysis for future cellular networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–6.
- [219] M. Alias, N. Saxena, and A. Roy, "Efficient cell outage detection in 5G HetNets using hidden Markov model," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 562–565, Mar. 2016.
- [220] J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Reading, MA, USA: Addison-Wesley, 1984.
- [221] H. Eckhardt, S. Klein, and M. Gruber, "Vertical antenna tilt optimization for LTE base stations," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, Yokohama, Japan, May 2011, pp. 1–5.
- [222] H. Peyvandi, A. Imran, M. A. Imran, and R. Tafazolli, "On Pareto-Koopmans efficiency for performance-driven optimisation in self-organising networks," in *Proc. IET Intell. Signal Process. Conf. (ISP)*, Dec. 2013, pp. 1–6.
- [223] H. Hu, J. Zhang, X. Zheng, Y. Yang, and P. Wu, "Self-configuration and self-optimization for LTE networks," *IEEE Commun. Mag.*, vol. 48, no. 2, pp. 94–100, Feb. 2010.
- [224] H.-M. Zimmermann, A. Seitz, and R. Halfmann, "Dynamic cell clustering in cellular multi-hop networks," in *Proc. 10th IEEE Singapore Int. Conf. Commun. Syst.*, 2006, pp. 1–5.

- [225] M. Al-Rawi, "A dynamic approach for cell range expansion in interference coordinated LTE-advanced heterogeneous networks," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, Singapore, 2012, pp. 533–537.
- [226] S. Tomforde, A. Ostrovsky, and J. Hähner, "Load-aware reconfiguration of LTE-antennas dynamic cell-phone network adaptation using organic network control," in *Proc. 11th Int. Conf. Informat. Control Autom. Robot. (ICINCO)*, vol. 1. Vienna, Austria, Sep. 2014, pp. 236–243.
- [227] L. Davis, *Handbook of Genetic Algorithms*. New York, NY, USA: Van Nostrand, 1991.
- [228] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *Computer*, vol. 27, no. 6, pp. 17–26, Jun. 1994.
- [229] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.
- [230] F. J. Mullany, L. T. W. Ho, L. G. Samuel, and H. Claussen, "Self-deployment, self-configuration: Critical future paradigms for wireless access networks," in *Proc. Workshop Auton. Commun.*, 2004, pp. 58–68.
- [231] O. G. Aliu, M. Mehta, M. A. Imran, A. Karandikar, and B. Evans, "A new cellular-automata-based fractional frequency reuse scheme," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1535–1547, Apr. 2015.
- [232] L. T. W. Ho, I. Ashraf, and H. Claussen, "Evolving femtocell coverage optimization algorithms using genetic programming," in *Proc. IEEE 20th Int. Symp. Pers. Indoor Mobile Radio Commun.*, Sep. 2009, pp. 2132–2136.
- [233] L. S. Mohjazi, M. A. Al-Qutayri, H. R. Barada, K. F. Poon, and R. M. Shubair, "Self-optimization of pilot power in enterprise femtocells using multi objective heuristic," *J. Comput. Netw. Commun.*, vol. 2012, 2012, Art. no. 303465.
- [234] A. Quintero and S. Pierre, "On the design of large-scale UMTS mobile networks using hybrid genetic algorithms," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2498–2508, Jul. 2008.
- [235] V. Capdevielle, A. Feki, and A. Fakhreddine, "Self-optimization of handover parameters in LTE networks," in *Proc. 11th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, May 2013, pp. 133–139.
- [236] L. Du, J. Bigham, L. Cuthbert, C. Parini, and P. Nahi, "Using dynamic sector antenna tilting control for load balancing in cellular mobile communications," in *Proc. Int. Conf. Telecommun. (ICT)*, vol. 2. 2002, pp. 344–348.
- [237] Z. Jiang, P. Yu, Y. Su, W. Li, and X. Qiu, "A cell outage compensation scheme based on immune algorithm in LTE networks," in *Proc. 15th Asia-Pac. Netw. Oper. Manag. Symp. (APNOMS)*, Hiroshima, Japan, Sep. 2013, pp. 1–6.
- [238] W. Li, P. Yu, M. Yin, and L. Meng, "A distributed cell outage compensation mechanism based on RS power adjustment in LTE networks," *China Commun.*, vol. 11, no. 13, pp. 40–47, 2014.
- [239] L. Xia, W. Li, H. Zhang, and Z. Wang, "A cell outage compensation mechanism in self-organizing RAN," in *Proc. 7th Int. Conf. Wireless Commun. Netw. Mobile Comput. (WiCOM)*, Wuhan, China, Sep. 2011, pp. 1–4.
- [240] J. Kittler, "Feature selection and extraction," *Handbook of Pattern Recognition and Image Processing*. Orlando, FL, USA: Academic Press, 1986, pp. 59–83.
- [241] F. Chernogorov, J. Turkka, T. Ristaniemi, and A. Averbuch, "Detection of sleeping cells in LTE networks using diffusion maps," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, Yokohama, Japan, May 2011, pp. 1–5.
- [242] I. K. Fodor, "A survey of dimension reduction techniques," Center Appl. Sci. Comput., Lawrence Livermore Nat. Lab., Tech. Rep. UCL-148494, pp. 1–18, 2002.
- [243] P. Cunningham, "Dimension reduction," in *Machine Learning Techniques for Multimedia*. Berlin, Germany: Springer, 2008, pp. 91–112.
- [244] R. R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmonic Anal.*, vol. 21, no. 1, pp. 5–30, 2006.
- [245] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [246] E. Bastuğ, M. Bennis, and M. Debbah, "A transfer learning approach for cache-enabled wireless networks," in *Proc. 13th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, May 2015, pp. 161–166.
- [247] W. Wang, J. Zhang, and Q. Zhang, "Transfer learning based diagnosis for configuration troubleshooting in self-organizing femtocell networks," in *Proc. IEEE Glob. Telecommun. Conf. (GLOBECOM)*, Houston, TX, USA, 2011, pp. 1–5.
- [248] A. Imran, E. Yaacoub, Z. Dawy, and A. Abu-Dayya, "Planning future cellular networks: A generic framework for performance quantification," in *Proc. 19th Eur. Wireless Conf. (EW)*, Guildford, U.K., Apr. 2013, pp. 1–7.
- [249] D. Kim, B. Shin, D. Hong, and J. Lim, "Self-configuration of neighbor cell list utilizing E-UTRAN NodeB scanning in LTE systems," in *Proc. 7th IEEE Consum. Commun. Netw. Conf.*, Las Vegas, NV, USA, 2010, pp. 1–5.
- [250] H. Sanneck, Y. Bouwen, and E. Troch, "Context based configuration management of plug & play LTE base stations," in *Proc. IEEE Netw. Oper. Manag. Symp. (NOMS)*, 2010, pp. 946–949.
- [251] A. Eisenblatter, U. Turke, and C. Schmelz, "Self-configuration in LTE radio networks: Automatic generation of eNodeB parameters," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, Budapest, Hungary, 2011, pp. 1–3.
- [252] D. Chen, J. Schuler, P. Wainio, and J. Salmelin, "5G self-optimizing wireless mesh backhaul," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPs)*, Hong Kong, Apr. 2015, pp. 23–24.
- [253] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.
- [254] B. Sas, K. Spaey, and C. Blondia, "A SON function for steering users in multi-layer LTE networks based on their mobility behaviour," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, Glasgow, U.K., May 2015, pp. 1–7.
- [255] B. Sas, K. Spaey, and C. Blondia, "Classifying users based on their mobility behaviour in LTE networks," in *Proc. 10th Int. Conf. Wireless Mobile Commun. (ICWMC)*, 2014, pp. 198–205.
- [256] C. Dhahri and T. Ohtsuki, "Cell selection for open-access femtocell networks: Learning in changing environment," *Phys. Commun.*, vol. 13, pp. 42–52, Dec. 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1874490714000445>
- [257] S. Samulevicius, T. B. Pedersen, and T. B. Sorensen, "MOST: Mobile broadband network optimization using planned spatio-temporal events," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, Glasgow, U.K., 2015, pp. 1–5.
- [258] L.-C. Wang, S.-H. Cheng, and A.-H. Tsai, "Bi-SON: Big-data self organizing network for energy efficient ultra-dense small cells," in *Proc. IEEE 84th Veh. Technol. Conf. (VTC Fall)*, Sep. 2016, pp. 1–5.
- [259] R. Barco, P. Lazaro, and P. Munoz, "A unified framework for self-healing in wireless networks," *IEEE Commun. Mag.*, vol. 50, no. 12, pp. 134–142, Dec. 2012.
- [260] S. Russell, P. Norvig, and A. Intelligence, "A modern approach," in *Artificial Intelligence*, vol. 25. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995, p. 27.
- [261] J. Wang, Y. Wu, N. Yen, S. Guo, and Z. Cheng, "Big data analytics for emergency communication networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1758–1778, 3rd Quart., 2016.
- [262] L. Bottou and Y. Bengio, "Convergence properties of the K-means algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, 1995, pp. 585–592.
- [263] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3, pp. 279–292, 1992. [Online]. Available: <http://dx.doi.org/10.1007/BF00992698>
- [264] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, vol. ASSP-3, no. 1, pp. 4–16, Jan. 1986.
- [265] G. A. Fink, *Markov Models for Pattern Recognition: From Theory to Applications*. London, U.K.: Springer, 2014.
- [266] V. D. Blondel, A. Decuyper, and G. Krings, "A survey of results on mobile phone datasets analysis," *EPJ Data Sci.*, vol. 4, no. 1, p. 10, 2015.
- [267] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.



Paulo Valente Klaine (S'17) received the B.Eng. (with Distinction) degree in electrical and electronic engineering from the Federal University of Technology–Paraná, Brazil, in 2014 and the M.Sc. (with Distinction) degree in mobile communications systems from the University of Surrey, Guildford, U.K., in 2015. He is currently pursuing the Ph.D. degree with the School of Engineering, University of Glasgow. He was with the 5G Innovation Centre, University of Surrey in 2016. His main interests include self organizing cellular networks and the application of machine learning algorithms in wireless networks.



Muhammad Ali Imran (M'03–SM'12) received the M.Sc. (Distinction) and Ph.D. degrees from Imperial College London, U.K., in 2002 and 2007, respectively. He is the Vice Dean Glasgow College UESTC and a Professor of Communication Systems with the School of Engineering, University of Glasgow. He is an Affiliate Professor with the University of Oklahoma, USA, and a Visiting Professor with 5G Innovation Centre, University of Surrey, U.K. He has over 18 years of combined academic and industry experience, working primarily in the research areas of cellular communication systems. He has been awarded 15 patents, has authored/co-authored over 300 journal and conference publications, and has been a Principal/Co-Principal Investigator on over six million in sponsored research grants and contracts. He has supervised over 30 successful Ph.D. graduates. He was a recipient of the Award of Excellence in Recognition of his academic achievements, conferred by the President of Pakistan, the IEEE Comsocs Fred Ellersick Award 2014, the FEPS Learning and Teaching Award 2014, and the Sentinel of Science Award 2016. He was twice nominated for Tony Jeans Inspirational Teaching Award. He is a shortlisted finalist for the Wharton-QS Stars Awards 2014, QS Stars Reimagine Education Award 2016 for innovative teaching and VCs learning, and Teaching Award in University of Surrey. He is a Senior Fellow of Higher Education Academy, U.K.



Richard Demo Souza (S'01–M'04–SM'12) was born in Florianópolis-SC, Brazil. He received the B.Sc. and D.Sc. degrees in electrical engineering from the Federal University of Santa Catarina (UFSC), Brazil, in 1999 and 2003, respectively. In 2003, he was a Visiting Researcher with the Department of Electrical and Computer Engineering, University of Delaware, USA. From 2004 to 2016, he was with the Federal University of Technology–Paraná, Brazil. Since 2017, he has been with UFSC, Brazil, where he is an Associate Professor. His research interests are in the areas of wireless communications and signal processing. He has served as an Associate Editor for the IEEE COMMUNICATIONS LETTERS, *EURASIP Journal on Wireless Communications and Networking*, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He was co-recipient of the 2014 IEEE/IFIP Wireless Days Conference Best Paper Award and the 2016 Research Award from the Cuban Academy of Sciences, the supervisor of the awarded Best Ph.D. Thesis in Electrical Engineering in Brazil in 2014. He is a Senior Member of the Brazilian Telecommunications Society.



Oluwakayode Onireti (S'11–M'13) received the B.Eng. (Hons.) degree in electrical engineering from the University of Ilorin, Ilorin, Nigeria, in 2005, the M.Sc. (Hons.) degree in mobile and satellite communications, and the Ph.D. degree in electronics engineering from the University of Surrey, Guildford, U.K., in 2009 and 2012, respectively. From 2013 to 2016, he was a Research Fellow with ICS/5GIC, University of Surrey. He is currently a Research Associate with the School of Engineering, University of Glasgow. His main research interests include

self-organizing cellular networks, energy efficiency, multiple-input multiple-output, and cooperative communications. He has been actively involved in projects such as ROCKET, EARTH, Greencom, QSON, and Energy proportional ENodeB for LTE-Advanced and Beyond. He is currently involved in the DARE project, a ESFRC funded project on distributed autonomous and resilient emergency management systems.