Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

Maestría en Telecomunicaciones

Análisis comparativo de algoritmos de aprendizaje máquina y su aplicación en sistemas de compensación de carga social-conscientes para redes móviles de última generación.

Comparative analysis of machine learning algorithms and their application in social-aware load balancing systems for next generation mobile networks

Trabajo de Titulación previa a la obtención del título de Magíster en Telecomunicaciones

AUTOR:

Ing. Milton Andrés León Bustamante

DIRECTOR:

Ing. Renato Benjamín Torres Carrión, Mg. Sc.

Loja – Ecuador

2023

Certificación

Loja, 25 de septiembre de 2023

Ing. Renato Benjamín Torres Carrión Mg. Sc.

DIRECTOR DE TRABAJO DE INVESTIGACIÓN

CERTIFICO:

Que he revisado y orientado todo el proceso de la elaboración del Trabajo de Titulación denominado: **Análisis comparativo de algoritmos de aprendizaje máquina y su aplicación en sistemas de compensación de carga social-conscientes para redes móviles de última generación**, de autoría del estudiante **Milton Andrés León Bustamante**, previa a la obtención del título de **Magíster en Telecomunicaciones**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja para el efecto, autorizo la presentación para la respectiva sustentación y defensa.

Firmado electrónicamente por:
RENATO BENJAMIN TORRES CARRION

Ing. Renato Torres Mg. Sc.

DIRECTOR DE TRABAJO DE INVESTIGACIÓN

Autoría

Yo, **Milton Andrés León Bustamante**, declaro ser autor del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación del Trabajo de Titulación en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:

Cédula de Identidad: 1105657645

Fecha: 25 de septiembre de 2023.

Correo electrónico: moltilion@live.com - moltilion@gmail.com

Teléfono: +36702874997

Carta de autorización por parte del autor para la consulta, reproducción parcial o total y publicación electrónica del texto completo del presente Trabajo de Titulación

Yo, **Milton Andrés León Bustamante**, declaro ser autor del Trabajo de TITULACIÓN denominado: **Análisis comparativo de algoritmos de aprendizaje máquina y su aplicación en sistemas de compensación de carga social-conscientes para redes móviles de última generación**, como requisito para optar por el título de **Magíster en Telecomunicaciones**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del trabajo de investigación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Budapest, a los 05 días del mes de agosto de dos mil veintitrés.

Firma: AndrésLeónB

Autor: Milton Andrés León Bustamante

Cédula: 1105657645

Dirección: Calle Dob 46A, Budapest, Hungría

Correo Electrónico: moltilion@live.com - moltilion@gmail.com

Teléfono: +36702874997

DATOS COMPLEMENTARIOS:

Director de Trabajo de Titulación: Ing. Renato Benjamín Torres Carrión, Mg. Sc.

Dedicatoria

A mis viejos y mis hermanos.

Lo más valioso de mi vida.

Milton Andrés León Bustamante

Agradecimiento

Quisiera agradecer a la Universidad Nacional de Loja por darme la oportunidad de pasar una vez más por sus aulas para fortalecer mis conocimientos técnicos y científicos sobre las telecomunicaciones.

Quisiera agradecer al claustro docente de la universidad y a los profesores invitados de la presente maestría por compartir sus conocimientos.

Agradezco a mis padres, por ser el soporte de mi vida,

Quiero agradecer de manera especial al Ing. Renato Benjamín Torres Carrión, Mg. Sc. por guiarme en el presente trabajo de tesis, que me ha permitido entender de mejor manera los conceptos de SON, self-optmization y balanceo de carga en redes de nueva generación.

Milton Andrés León Bustamante

Table of Contents

Index of Tables

Index of Figures

1 Title

“Comparative analysis of machine learning algorithms and their application in social-aware load balancing systems for next generation mobile networks”

2 Summary

El balanceo de carga móvil (MLB) es un caso de uso de las Redes Autoorganizadas (SON) cuyo objetivo es el de mantener el buen desempeño de la red al evitar el aparecimiento de celdas sobrecargadas, las cuales pueden llevar a un empobrecimiento de la experiencia de usuario debido a la falta de recursos. Un habilitador clave para la implementación del MLB ha sido el aprendizaje máquina (ML), ya que permite que la red aprenda cuál es la mejor configuración que puede ser adoptada en el caso de la ocurrencia de una celda sobrecargada y converger a una solución aceptable. Sin embargo, en los últimos años, un nuevo habilitador que puede impulsar el desempeño de los algoritmos de ML ha aparecido: los sistemas social conscientes. En el presente trabajo, se realizó una revisión de literatura sistemática para explorar propuestas actuales de algoritmos de MLB basados en ML que emplean información social consciente para su operación. El objetivo es identificar sus características principales, y basados en ellas, determinar sus posibles escenarios de aplicación. Finalmente, se modeló un escenario urbano de prueba, y se realizó un análisis cualitativo, usando las características definidas, para definir cuál algoritmo sería más conveniente de ser implementado en el escenario de prueba.

Palabras Clave: redes celulares, redes de nueva generación, SON, balanceo de carga, machine learning, social consientes, context-aware, optimización proactiva de la red.

2.1 Abstract

Mobile load balancing (MLB) is a use case of Self-Organizing Networks (SON) whose objective is to maintain the good performance of the network by avoiding the appearance of overloaded cells, which can lead to a decrease in user experience due to the lack of resources. A key enabler for the implementation of MLB has been machine learning (ML), as it allows the network to learn what is the best configuration that can be adopted in case a load hotspot occurs and converge to an acceptable solution. However, in the past years, a new enabler that can boost the performance of ML algorithms has appeared: social-aware systems. In the hereby work, a systematic literature review is performed to explore current proposals of Machine Learning-based MLB algorithms that employ social-aware information for their operation. The aim is to identify their main characteristics, and based on them, determine their possible application scenarios. Finally, an urban test scenario is modeled, and a qualitative analysis is performed, using the defined characteristics, to define which algorithm would be more suitable to be implemented in the test scenario.

Keywords: cellular networks, next generation networks, SON, load balancing, machine learning, social-aware, context-aware, proactive network optimization.

3 Introduction

Next generation mobile networks (NGMN) have experienced an important raise in complexity during recent years, promoted by the need of providing higher network capacities with stringent requirements for satisfying the demands of a continuously increasing number of connected devices, each of them with different Quality of Service (QoS) needs.

This increase in complexity comes at the cost of a higher operational expenditure (OPEX) and more difficult network optimization [1]. Self-Organizing Networks (SON) is a network automation concept that seeks to bring intelligence to the network by giving it the capabilities of self-configuration, self-optimization, and self-healing. The goals of SON are to simplify network management, to reduce capital and operational costs (CAPEX/OPEX), and to improve the performance of the network.

Self-optimization in SON refers to the application of techniques for optimizing the network parameters in an automatic manner. This process is carried out in the following manner: 1) The network collects data from its operation through measurements, 2) new operation parameters are computed based on the evaluation of the measurements and finally, 3) the new parameter values are implemented in the network. [2] One of the most important use cases of self-optimization is Mobile Load Balancing (MLB).

Mobile Load Balancing (MLB) is a self-optimization method for managing cell congestion by transferring traffic from a congested cell to adjacent cells with available resources. The main goal is to maintain the end-user experience and a good performance of the network. Mobile load balancing is implemented by measuring several KPIs from the network and the UEs, for using them to tune the parameters that govern the handover (HO) process. The handover process is the responsible for transferring a UE from one cell to another. Thus, when a cell is overloaded, the network sets its HO parameters in such a way that allows an adjacent cell to take some of the users of the congested cell.

Mobile networks generate an enormous amount of data that can enable the making of better management decisions. Machine learning (ML) algorithms are one of the most popular methods to analyze network data for fulfilling the objectives of SON. In literature, it is possible to find many approaches that use ML algorithms for solving the load balancing problem; some of the most commonly implemented methods for this are reinforcement learning, fuzzy controllers, and regression trees.

In NGMN, traffic is regarded as fast-changing and with an imbalanced spatial-temporal distribution, which poses a challenge for load balancing algorithms [3]. This is where the concept of context-awareness (CA) comes into play. Context is any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object [4]. A context-aware computing system is capable of collecting and analyzing the context information for adjusting its operation accordingly. For instance, it is possible to characterize the traffic distribution of a mobile network by understanding the user's behavior and its geolocation.

The context is often challenging to uncover, unfold over time, and it is difficult to collect personal data due to privacy concerns [1]. However, through the social interaction of the users with the network and with their pairs is possible to deduce the traffic context. For this reason, public online data is used to retrieve user behavior information. Online data can come from a wide variety of sources, including social networks, video/photo sharing sites, online forums, product reviews/ratings, and wikis [1].

As the number of users and requirements keeps growing, networks will have to increase their complexity to keep up with the demand. The potential of coupling ML with social-aware systems constitutes a viable solution to enable the load optimization of mobile networks in an automated and efficient manner. This survey seeks to explore current proposals of ML-based social-aware load balancing and their viability to be implemented in a real-life scenario.

The rest of this work is organized as follows: Section 4 discusses similar survey works in which the state of the art of SON, self-optimization, and load balancing methods is reviewed. The main trends in MLB are identified here. Section 5 begins by describing the methodology followed for doing the systematic literature review. It then introduces the social-aware load balancing algorithms that were identified in the literature review and explains their operation. In the same section, an urban test case scenario is modeled; this scenario will be used to compare the MLB algorithms. In Section 6, the algorithms are characterized and compared to find out which one would be more suitable to be applied to the test deployment. The results of this comparison are shown here. Next, in Section 7, the obtained results are discussed, in Section 8 the conclusions reached during the execution of this work are stated and finally, in Section 9, some recommendations are shared.

4 Theoretical Framework

4.1 Related Works

There have been several survey studies regarding the different approaches taken for solving the load balancing problem in SON. In this section, we will discuss the most important trends in SON, self-optimization, and load balancing that we were able to identify from the review of a group of surveys conducted by different authors during the 2017 to 2022 period. A comparison between the methodologies and the results of these surveys is shown in Table 1. At the end of this section, we will focus on the works related to the use of social-aware data for load balancing.

Table 1. Comparison of survey studies on SON, self-optimization, and load balancing.

Study	Publication Date	Study Description	Findings
[5]	2017	A literature review of the past 15 years until the year of the study was performed. The authors classified the reviewed works in terms of the used learning method, the tackled SON use case, and how each solution performed.	The study built a foundation in which researchers can understand the basics of the most important ML algorithms and how they are applied to different SON use cases. It is stressed that ML solutions are necessary for enabling autonomous and intelligent networks.
[6]	2018	The review focuses on SON for 5G networks and the importance of ML for enabling SON. It introduces the basic concepts of SON and 5G network management. Then, it focuses on the evolution of SON in the 3GPP standard. it also provides guidelines for selecting the most appropriate ML method for each use case. Finally, the main sources of data for intelligent network management are explored.	The study showed that NG-SON has to be redefined as an embedded feature of 5G network, instead of only an add-on. It also highlighted the fact that current networks already generate enormous amounts of data, that when properly processed, can allow autonomous and intelligent management of the network.
[1]	2020	The study is divided into two parts. In the first part, a review of the methods for inferring the CPSS context from heterogeneous data sources is presented. In the second part, the study explored the methods used for integrating context knowledge with proactive optimization techniques. Both sections present state-of-the-art ML techniques for performing these tasks.	The study showed the most important technologies for online-data analytics that can support the paradigm change from reactive optimization to proactive optimization. The study also created a foundation from which researchers can understand what are the most valuable context data and their sources.
[7]	2021	The study is a detailed survey of SON evolution from 4G to 5G	The study presented an exhaustive overview of the SON paradigm. It

		networks. It begins by introducing the comprehensive background of SON. Next, it introduces the ML solutions implemented by SON. Finally, it explores open issues and research trends.	also showed the challenges that SON has to overcome for its implementation in 5G networks and stressed the necessity of ML methods for empowering SON to meet 5G requirements. This work constitutes a valuable step for opening research lines about SON.
[8]	2022	This study presents a road map for introducing cost-effective, flexible, and intelligent load balancing in HetNets. The study starts by providing an overview of the load balancing problem. Next, it presents a review of ML-based load balancing models. Finally, a summary of the challenges and future research lines are presented.	The study presented a solid report on the development of ML-based load balancing methods in HetNets, the technical issues relating to their implementation, their performance, and their shortcomings.

4.1.1 Machine Learning

The increasing number of connected devices, the appearance of new services, and the demand for higher capacity have raised the complexity of mobile networks. It is possible to see this in the enormous amounts of tunable parameters that nodes possess for adjusting their operation. In addition, the appearance of heterogeneous networks (HetNets) will only increase the number of connected nodes. In this scenario, network management becomes a difficult task with a high cost. Therefore, the need to automatize the operation and optimization of the network is justified.

Huge amounts of data are generated from the control and management functions of the networks [6]. With the right tools to analyze this data, it will be possible to enable intelligence, self-awareness, and self-adaptability, thus reducing the complexity of network management. Machine Learning and Big Data are technologies that have the potential to empower the intelligent operations of SON [7].

ML is a set of methods that enable computers to learn, adapt and optimize a model for pattern recognition [8]. The attractive part of ML methods lies in their ability to learn from the data generated by the system, allowing it to forecast work scenarios and adapt accordingly. This improves the performance of the network and reduces OPEX by limiting human intervention in network management. At this point, there exist many optimization works focusing on exploiting Big Data, building a SON engine architecture, and developments in machine learning [1].

Most of the works that seek to bring SON solutions to 5G are based on ML [7], leaving classical optimization methods behind. For example, in [9], a three-fold approach for load balancing in ultra-dense networks (UDNs) is proposed. The authors introduce a two-layers architecture where the upper layer implements a k-means algorithm for clustering cells to adapt to their local traffic variations. The bottom layer uses a deep reinforcement learning (DRL) algorithm for learning the best MLB policy for intra-cluster load balancing. Finally, an offline evaluation system is applied for ensuring that the system can always operate with the optimal MLB policy and enable the exploration of policies beyond the ones currently implemented. Likewise, in [10] the authors presented a mobility-aware DRL solution for load balancing. Their method configures the HO parameters following the mobility patterns of the users to achieve approximately the same quality level for each of them. The results show a decrease in the number of unsatisfied subscribers and a more balanced network.

Classical optimization methods such as genetic algorithms, linear programming, particle swarm algorithms, or other heuristic approaches, can be applied together with machine learning mechanisms as the optimization search strategy of the mobile network [11]. Examples of this can be found in [12] and [13], where RL methods are combined with fuzzy logic controllers (FLC), which are a classical method of load balancing, for optimizing the performance of the latter. These works demonstrated that it is possible to obtain optimization strategies that combine the best of both approaches, the fast response of FLCs, and the improvement in performance due to the RL algorithms [12]. The use of ML algorithms for optimizing classical load balancing methods can enable a self-adaptative, optimal, and cost-effective optimization in scenarios where computational resources are limited.

4.1.2 Proactive Load Balancing

In order to meet the requirements of 5G and NGMN, current classical SON methods have to evolve from reactive to proactive [7]. Classical SON methods are reactive, which means that they get into action after a problem occurs and is detected, whereas, in proactive SON, the network can forecast operating scenarios and tune its parameters for adapting to changes in traffic before a problem appears. This is important because the time-scale in which an optimization algorithm operates heavily influences the QoS [1]. A long delay in the optimization answer when a problem arises, will lead to user dissatisfaction. For this reason, the tendency in optimization algorithms is to push their time-scale to be near real-time and even proactive [1].

For enabling the change of paradigm from reactive optimization to proactive optimization, the network must be able to identify the context in which it operates. The context can be derived from data such as geolocation, public online data of attendance to events, and historical network usage. For example, Klaine et al. [5] explain that through the analysis of historical data, it is possible to allow the construction of normal network operation scenarios. By knowing what are the parameters of regular network operation, it is possible to forecast where a possible issue might arise.

4.1.3 Social-Awareness

Network optimization can be greatly improved with the understanding of user behavior and the spatial-temporal distribution of traffic [1]. The user information from which we can derivate this data constitutes the *context* of the network. Examples of this kind of information include the geolocation of the UEs, the popularity of the content consumed by the users, the user’s sentiment, and the relationships between different users. The computer systems that use the context for taking decisions, executing tasks, and delivering services are called *context-aware*.

It is challenging to elucidate the context of a mobile network. User data is difficult to retrieve due to privacy concerns. Fortunately, internet users and more specifically, social media users, generate a great amount of data voluntarily. This data contains information that strongly correlates to the behavior of the user, making it an optimal source for inferring their context. Social networks are one of the main sources of context data. Examples of these platforms are Twitter, Facebook, or Instagram. Other sources of context data include calendars, open databases, event aggregators, photo/video sharing sites, online forums, product review ratings, and wikis [1] [14]. In this work, the term context-aware will be used as a generalization of the paradigm of social-aware information, which includes information such as location, velocity and public social network data.

From the surveys analyzed for this section, only the work of Bo Ma et al. [1] made a review specifically focused on context-aware balancing methods. Another survey that touched on this approach, although superficially, was Gures et al [8], while the others did not mention it at all. The survey of [1] presented many interesting context-aware load balancing solutions such as the one offered by [15], where the authors used Twitter for geolocating the occurrence of events, such as festivals. For this, they divided the region they wanted to monitor into smaller regions of interest inside of which they can establish the pattern of crowds. Next, they established the regularity of crowd behaviors in the regions of interest using historical geo-tagged tweets and finally, they detected the appearance of events in a test data set by comparing it with the established regularity. Another interesting work is the one made by [16], where the popularity of YouTube videos is predicted for content-aware proactive caching, as popular content generates a big portion of the load in the network. For predicting the popularity of videos, a set of features are extracted from the videos and stored in a vector. Prediction models are then applied to these features. The goal of proactive caching is to alleviate the load stress of the network by storing frequently-accessed data in base stations (BSs) near the users.

5 Methodology

5.1 Systematic Literature Review

The Systematic Literature Review (SLR) was performed following the methodology established by [17]. The review begins with the planning stage in which we want to understand the current state of the research problem. For this, a set of research questions was defined to properly characterize the research problem we wish to explore with our study. These questions are:

1 What machine learning algorithm was used for the load balancing process?
2 In which type of geographical and use case scenario (urban, rural) is the network operating?
3 What social data was used for the load balancing process?
4 What parameters were optimized by the ML algorithm?

The second stage is the development of a review protocol where a set of inclusion and exclusion criteria for the review is established, as well as parameters for organizing the results of the review. The inclusion criteria for this work are:

- Only papers published between 2015 and 2022 will be considered.
- Only papers that include the words machine learning, mobile load balancing, context-aware, or social-aware in the title, keywords or body text will be included.

On the other hand, the exclusion criteria are:

- Papers dealing with MRO, CCO, self-configuration, and self-healing will not be taken into account. These are concepts of SON related to load balancing but are not synonymous.
- Papers outside of the publication range defined in the inclusion criteria will not be considered.
- Documents that are not scientific papers.
- Papers outside of the field of computer science and engineering.

The final stage of the SLR is conducting the review, taking into consideration the review protocol. The most relevant works found using this method are presented in the next subsection.

5.2 Reviewed Works

5.2.1 Mobility Management Based on Reinforcement Learning

In 2015, Simsek et al. [18] proposed two approaches for Mobility Management (MM) and a context-aware UE scheduling method for HetNets, based on the use of reinforcement learning algorithms. The authors seek to provide both a long and short-term solution for the optimization of the HO parameters between macro and picocells. Their contributions can be described as follows:

- They proposed a set of MM methods that focus on long-term and short-term optimization solutions for HetNets. As a long-term solution, two reinforcement learning load balancing approaches are proposed, while a UE scheduling method is used as a short-term solution.
- The two long-term load balancing methods are first, a Multi-Armed Bandit (MAB) learning approach and second, a satisfaction-based learning approach.
- The short-term solution is implemented through a context-aware scheduler that employs the throughput history and the velocity of the UE as context data.

The objective of these solutions is to maximize the rate of the network.

The context-aware scheduler for short-term MM operates in the following manner: For each Resource Block (RB), a UE is chosen to be served by a BS in a given RB. The candidate UEs are sorted by their velocity, choosing first the UE with the slowest velocity in order to avoid favoring high-velocity UEs over slow ones [18]. For preventing the scheduler from allocating most of its resources to a newly handed over UE, the macro and picocells communicate between them through the X2 interface, so that when a UE is handed over from a microcell to a picocell, its rate history is provided to the picocell in terms of average rate.

On the other hand, both of the long-term MM solutions are implemented by considering a game in which the BSs are the players and there is a set of actions they can take at every optimization step. In addition, each player has a utility function. The players learn how to optimize their load by following the next steps:

1. An action is chosen based on the utility function, which takes into account the total rate of the player at that moment.
2. The strategy for choosing the action is updated based on one of the two reinforcement learning algorithms that are proposed by the authors.
3. The newly handed-over UE is given a RB taking into account its velocity, average rate, and instantaneous rate in accordance with the context-aware scheduler described before.

The first reinforcement learning algorithm for long-term MM is the MAB. The elements used for the operation of this algorithm are as follows:

- The players are the macrocells and the picocells of the network.
- The actions are a set of Cell Range Expansion (CRE) bias values in dB that the macro and picocells can adopt.
- The utility function of the algorithm is a decision function composed of a term representing the total rate of a player and a term that takes into consideration the number of times an action has been chosen until that moment.

Considering all these elements, the MAB operates in the following manner:

1. In the first optimization step, a player selects each action in a random manner for initializing the learning process by receiving a reward for each action.
2. After the initialization, the player selects an action that maximizes the decision function.
3. Next, the player updates the parameters of the decision function based on the change in its cumulative reward produced by the chosen action.
4. Repeat step 2.

The second algorithm for MM optimization is a satisfaction-based learning approach. These approaches seek to guarantee the satisfaction of the players in a system [19]. In this case, the player is considered to be satisfied if it reaches a minimum level of total rate and if at least 90% of the UEs in the cell receive a certain average rate.

In this algorithm, the set of players and actions are the same as in the MAB approach. The utility function is the load of the cell. The actions are selected according to a probability distribution in the following manner:

1. During the first learning iteration, the probability of each action is equal and an action is chosen at random.
2. After the initialization iteration, the BS changes its action selection strategy only if the received utility does not meet the satisfaction criteria.
3. If the satisfaction condition is not reached, the player chooses an action based on the probability distribution.
4. A reward is given to the players based on the selected actions.
5. Finally, the probability of the chosen action is updated using a linear reward-inaction scheme.

System level simulations were carried for assessing the performance of the proposed MM solutions. The results show that the proposed approaches improve the performance of the network compared to traditional solutions. With these methods, the average UE throughput is increased by 80% and the handover failure is reduced by a factor of three [18].

5.2.2 A Markov Chains Model for the Development of Context-Aware HO Policies

In 2015, Guidolin et al. [20] developed a model based on a discrete-time Markov Chain (MC) to characterize the HO process of a mobile user and employ it as a base to derive optimal context-aware HO policies for increasing the performance of the user. In their approach, the authors begin by introducing a model that describes the operation of a UE along its moving trajectory inside a representative HetNet scenario. This model is then used to derive an expression for the average performance of the UE as a function of the HO parameters and other context parameters such as the velocity of the UE, the power of the macro and femtocells, the load of the cells, and the channel model. Finally, the model is used for deriving a context-aware HO policy (CAHP) for optimizing the HO parameters with respect to the gathered context information.

The considered HetNet model consists of a macro-BS (M-BS) and a femto-BS (F-BS) separated by a certain distance and that use the same frequency band. Despite the simplicity of this approach, it is capable of describing the most fundamental aspects of the HO process in HetNets. Furthermore, it can be generalized to more complex scenarios. In this configuration, the UE moves in a straight trajectory from the M-BS to the F-BS at a constant speed.

The propagation model used is the path-loss plus fading propagation model. It has to be noted that fading can cause an improper triggering of the HO process, producing the ping-pong effect.

The authors considered that the HO process is executed as modeled by the 3rd Generation Partnership Project (3GPP). The HO is started by the UE, which is periodically measuring the Reference Signal Received Power (RSRP) from the surrounding cells. When the UE detects that the difference between the RSRP of the serving and target cells drops below a certain HO hysteresis value λ_{th}, a timer known as Time-to-Trigger (TTT) is initialized to a certain value T and the countdown begins. If the RSRP difference comes back to be higher than the HO hysteresis threshold, then the countdown stops and the HO is aborted. On the other hand, if the difference remains below the threshold for the complete duration of the interval T, then the UE disconnects from the serving cell and connects to the target cell. This is known as the A3 event [21].

The next step in the modeling of the HO process is to determine an expression for the mean trajectory performance of the UE while it is moving from a M-BS to a F-BS. This derivation starts by considering that at any point of its trajectory, the connection state S of the UE can be defined as M, F, or H depending if the UE is connected to the M-BS, the F-BS, or if is temporarily disconnected because is executing the HO process. The obtained mean trajectory performance expression is proportional to the connection state

S at any point of the trajectory and to the performance experienced by the UE. In this work, the performance experienced by the UE is expressed in terms of the Shannon capacity, however, it can be generalized to use different performance metrics such as the HO failure rate or the ping-pong rate. During the HO process, the UE experiences many connection costs related to signaling. Because of this, it is assumed that during the HO the UE experiences zero capacity.

The computation of the mean trajectory performance is complex due to the fading process. The solution for this issue was to replace the continuous time model with a slotted-time model. In this way, the trajectory of the UE is analyzed at periods separated by the fading coherence time T_c. From the period of every time slot, it is possible to obtain the spatial granularity of the model.

The HO process is modeled as a non-homogeneous discrete-time Markov chain (MC). The modeling starts by denoting as N_T and N_H the number of space slots traveled by the UE during times T and T_H respectively. The states that represent the connection of the UE to either a M-BS or F-BS are denoted by the sets M_j and F_j respectively, where $j \in \{0, \ldots, N_T\}$. In the same manner, the states that represent the HO process of the UE are represented by the sets H_j and $\hat{H}_j$, in which $j \in \{1, \ldots, N_H\}$. The set H_j represents the handover from a macro to a femtocell, while the set $\hat{H}_j$ represents the handover from a femto to a macro cell.

When the UE is connected to a M-BS and enters the TTT period, it enters to the set of states M_j. The UE will evolve from the state M_j to M_{j+1} while the RSRP difference is below λ_{th}, otherwise, the TTT timer will reset and the UE will come back to the state M_0. On the other hand, if the RSRP condition is maintained until the state M_{NT} is reached, the UE will initialize the HO process and will enter the H_1 state. All the steps of the HO will be deterministically passed until the final state H_{NH}. After this, the UE will connect to the F-BS and it will enter to the state F_0, which means that the HO process has been successful. The evolution of the MC for switching from the F-BS to the M-BS is conceptually the same. Figure 1 depicts the MC model that describes the HO process.

If the target cell is partially loaded, then a HO based only in the RSPR may yield poorer results than simply staying in the current cell due to the traffic load. To account for the traffic load, the RSRP-based HO procedure described previously can be maintained, while the Cell Individual Offset (CIO) can be modified to account for different traffic loads. In practice, this translates to defining a new threshold $\lambda_{th}^{S,load}$ that is dependent on the load of both the M-BS and F-BS, respectively.

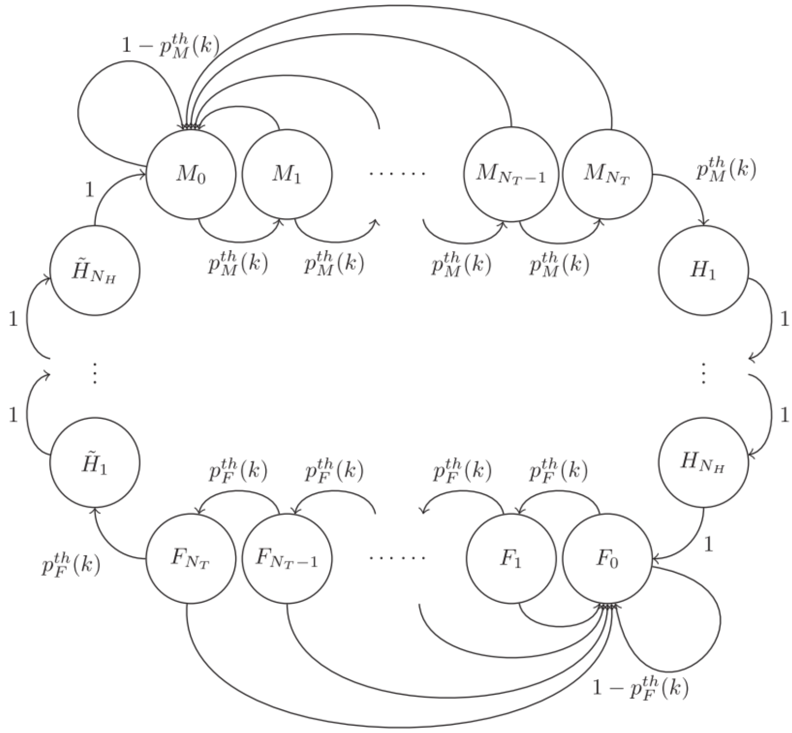

Figure 1. Discrete-time MC for modeling the HO process of a UE with arbitrary N_T and N_H. Taken from [20].

The threshold choice determines the characteristics of the load-aware HO algorithm [20]. A good approach for determining this threshold would be to adapt it in such a way that the relative performance gain experienced by the UE, when changing cells, remains constant.

The developed mathematical model can be used to derive a context-aware HO policy (CAHP). The context data used by the model are the transmit power of the cells, the path loss coefficients, the inter-BS distance, the carrier frequency, the UE velocity, and the load of the cells. The idea of the CAHP is to find the optimal TTT value that maximizes the average performance of the UE when crossing from one cell to another. The TTT value depends on the current context parameters, which are assumed to be known by the

UE. Pilot signals can carry some of the needed context information such as the cell load conditions, while others can be obtained from the UE itself, like the UE speed, which can be gathered from the GPS of the device.

In order to find the best TTT for a certain set of context information, the authors explored the change in the capacity experienced by the UE as they varied the context parameters. With this approach, they found that for low speeds of the UE, larger TTT periods are needed to avoid the ping-pong effect. In contrast, for very high UE speeds, the best policy is to avoid the HO completely, since the loss of capacity due to the signaling of the UE is not balanced by the capacity gain of connecting to the F-BS. Now, the speed threshold over which the best policy is to skip the HO depends on the size of the F-BS. For large cells, the HO losses are balanced by the gain in capacity obtained by connecting to the F-BS. For lower speeds, the optimal TTT depends on the UE speed but its independent of the size of the cells.

These conclusions were implemented in a CAHP algorithm that adapts the TTT of the UE in accordance with the measured context values. To evaluate the performance of this algorithm, a series of simulations that compared the results yielded by this method against the results given by fixed TTT policies were executed. From these simulations, it is evident that the CAHP outperforms the results of the fixed TTT policies and demonstrates that context-awareness can improve significantly the HO process.

5.2.3 A Framework for Context-Aware Self-Optimization

In 2016, Aguilar-Garcia et al. [22], proposed a framework for context-aware self-optimization (CA Self-Optimization) whit the objectives of 1) increasing the number of satisfied users despite the changes in the radio conditions of the cellular network and 2) converging to an optimal network status in the shortest time possible. Next, the authors also describe a load balancing use case based on the proposed framework and then proceeded to validate it through a simulation.

The CA Self-Optimization framework is based on the inclusion of a Context-Aware Module (CAM) for enhancing the capabilities of traditional self-optimization SON algorithms, by providing them with context information, in addition to traditional network

KPIs. The CAM is able to collect context information from multiple data sources including personal devices, location systems, social networks, image/video, inputs by manual users, and other data sources. Depending on the type of data used, the CA self-optimization algorithm can run at various time intervals, including 1 h, 30 min, 15 min, 5 min, 1 min, and can even allow the making of predictions about the status of the network.

The CA Self-Optimization framework is used to enhance a load balancing algorithm based on a FLC. The block diagram of a FLC is shown in Figure 2. The authors explored two load balancing algorithms based on FLCs: the Power Traffic Sharing (PTS) algorithm, and the Power Load Sharing (PLS) algorithm. These algorithms work in the following manner:

- PTS: In the first method, the FLC inputs are: 1) the difference between the cell's blocked calls and the average of the neighboring cells' blocked calls and 2) the FLC's own output used as feedback. The FLC optimizes the load sharing between the cells by adjusting their transmission power. This algorithm is run periodically per small cell.
- PLS: The inputs of this algorithm are 1) the number of slots used per Physical Resource Block (PRB) over the total available slots per PRB and 2) the feedback of the FLC. Its output is the same as the previous algorithm. This method is run periodically per small cell.

The variation in the transmission power produced by these methods is limited to the small-cell power range, i.e., from 0.016 mW to 250 mW.

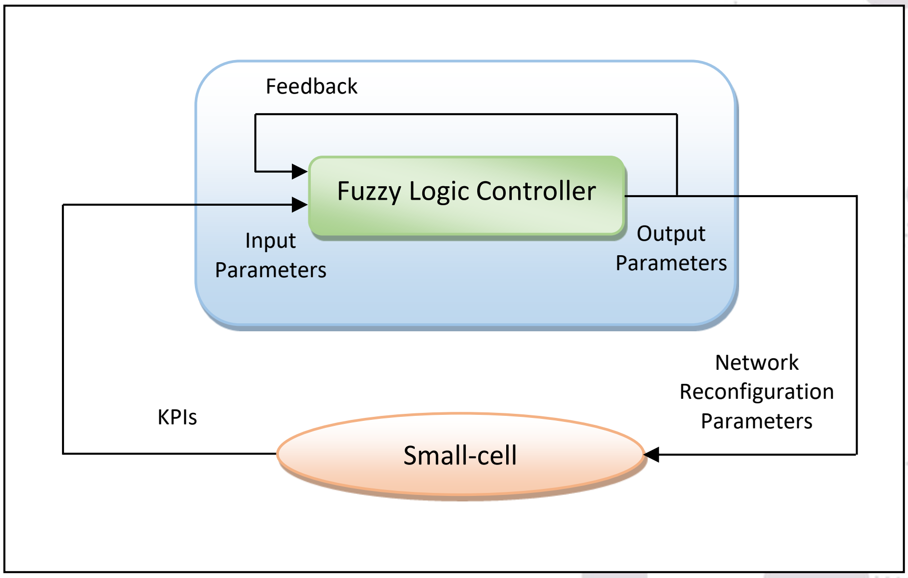

Figure 2. The general fuzzy logic controller-based load balancing algorithm. Taken from [22].

For the integration of the CAM with the load balancing algorithm, both processes work as separate modules. The CAM collects and classifies the context information and the FLC provides an optimization action based on the KPIs of the network. The combination of the outputs of each module is done through the implementation of a new module called Integration Module (IM). The IM analyzes the context information and decides if the output of the load balancing algorithm is applied to the network or modified. The integration of the load balancing (LB) and CA modules is depicted in Figure 3. The IM structure comprises five modules: a filter, a prediction database, an analyzer, a decision maker, and an acceleration function. Figure 4 illustrates this structure.

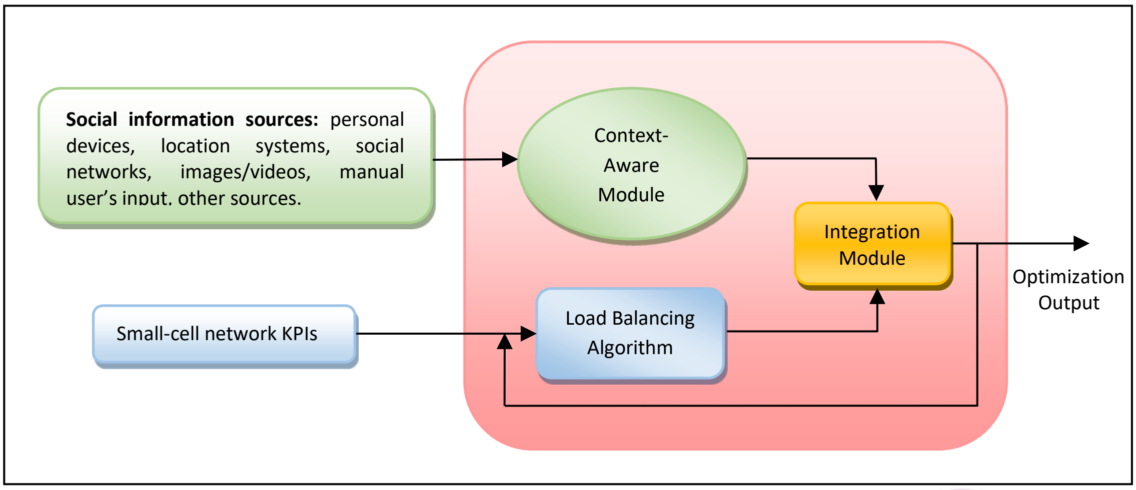

Figure 3. Integration scheme of the LB algorithm and the CA module. Taken from [22].

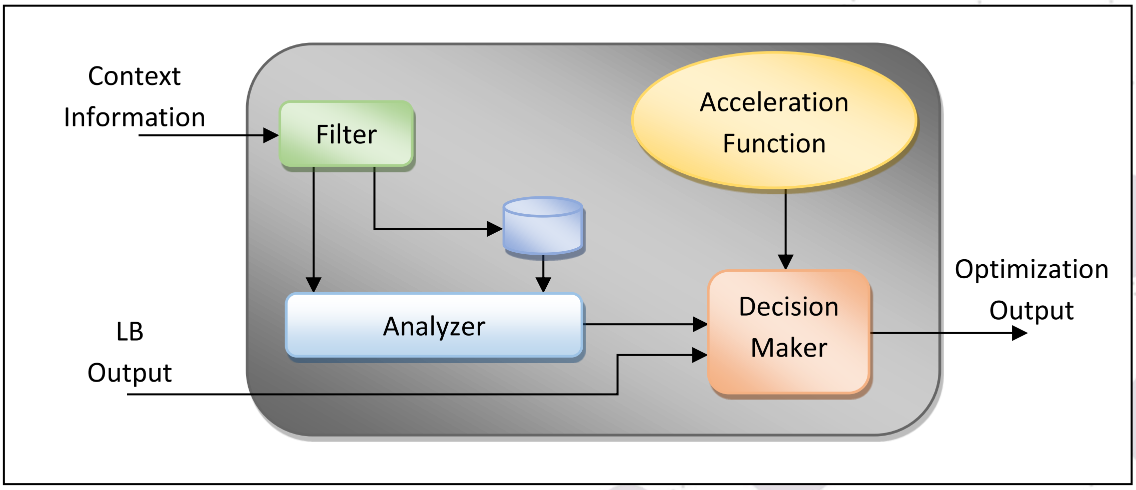

Figure 4. Architecture of the IM. Taken from [22].

The filter selects the context information relevant to the chosen load balancing method and eliminates non-useful data. This data is forwarded to the analyzer, while data that can provide information about future events is stored in the prediction database for future use. The analyzer determines if the evaluated cell is congested and if there is a high concentration of users close to the edge of the cell. When there is a similar received power from the serving cell and another neighboring cell, it can be deduced that a user is located at the cell border [22]. The outputs of the analyzer are the following:

- When the evaluated cell is a congested cell or the nearest neighbor of a congested cell, then the output value is 1.
- In the case that the cell is a neighboring cell of a congested cell, the output value is 0.
- Otherwise, the output is set to -1.

Both the output of the analyzer and the acceleration function is fed to the decision maker for determining the output of the CA load balancing algorithm. The acceleration function multiplies the LB algorithm output by a certain factor. In the case proposed by the authors, the acceleration function is a step function. According to the outputs of the analyzer, the outputs of the decision maker are as follows:

- 1: the output of the LB algorithm is doubled.
- 0: no variation is applied to the output of the LB algorithm.
- -1: the LB algorithm output is canceled.

The proposed load balancing method was tested using the dynamic system-level LTE simulator described in [23]. The results of the CA self-optimization show an important improvement in performance relative to the traditional load balancing methods. The convergence of the algorithm is reduced by approximately 50%, and the number of dissatisfied users is reduced by improving the UDR indicator.

5.2.4 Context-Aware-Driven Proactive Load Balancing

In 2020, Ma et al. [3] presented a context-aware proactive load balancing method for optimizing the load share of cellular networks in urban areas. This work made contributions in three aspects of the load optimization process: first, they introduce a method for detecting hotspots in the network caused by massive assistance events, with the use of Twitter data. Second, a load balancing method that employs the data of the hotspot-detecting method is described. Finally, this load balancing method is optimized with the introduction of an algorithm for establishing the best activation time of the LB method.

Before discussing the mentioned contributions in detail, the authors explained the operation framework of their context-aware proactive load balancing method. The framework is composed of two functional blocks, data sources and context-aware proactive network optimization. The data sources contain the necessary data for the operation of the load balancing method, whereas the optimization block is divided into four functions: social data collection, social data filtering, 3-stage data analytics, and proactive optimization. This framework is visualized in Figure 5.

The objective of the social data collection function is to capture raw tweets from the internet to pre-process them and produce a formatted data set. Next, the social data filtering module is designed to eliminate irrelevant data and to provide an appropriate numeric expression of this information. With this data, it is possible to forecast the occurrence of traffic hotspots due to the appearance of social events in the network. A 3-stage data analytics function that employs machine learning and statistical methods is used for extracting the context of the data. The first stage of this function produces a spatial traffic pattern by dividing the urban area into various Regions of Interest (ROI), for modeling the traffic in each of them. The division of the ROIs is made with the use of the Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. The second stage is the hotspot detection, which is implemented by counting the Tweets in the pixels of each ROI; the areas with a high count of Tweets are defined as hotspots. The traffic distribution in the ROIs is not static, but it changes with time. The last stage, the anomaly detection, aims to forecast the time and location of new hotspots when social events occur. This is done in two steps: 1) By modeling the regularity of the traffic with a training data set and 2) by finding outliers in the modeled regularity. In this work, the number of Tweets per hour is used as the indicator of traffic changes in the network.

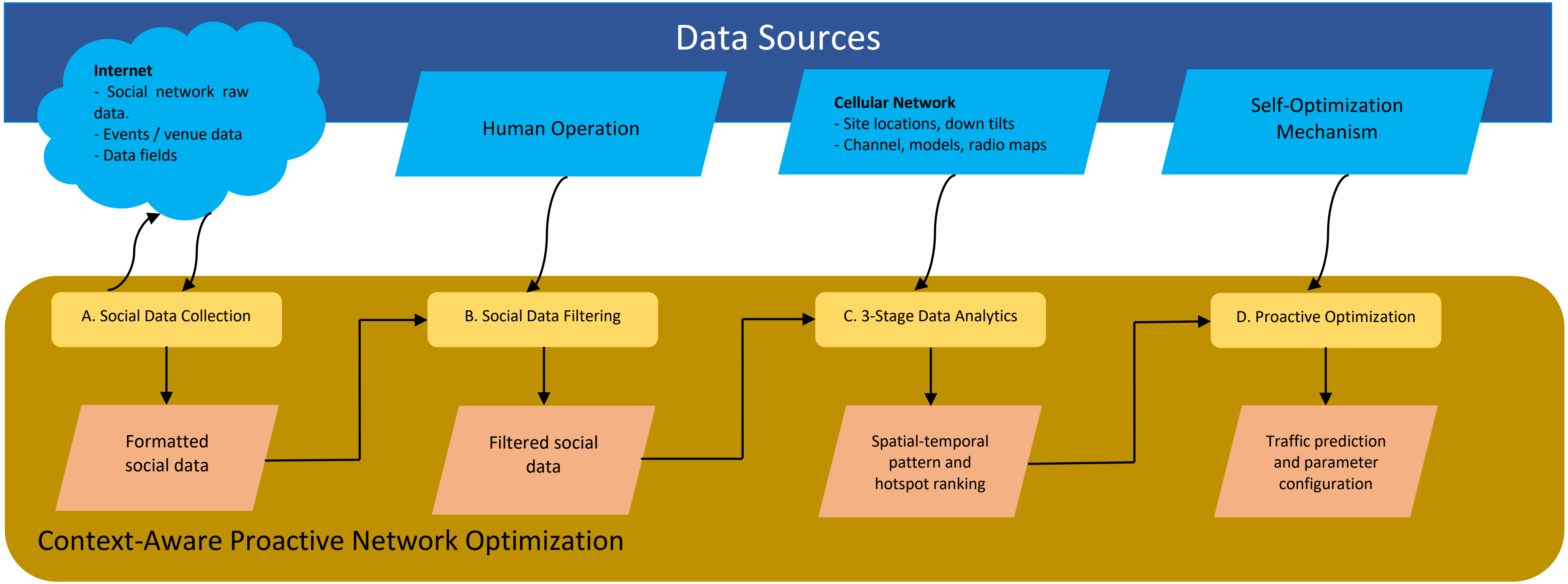

Figure 5. The framework of the proactive network optimization. Taken from [3].

The proactive optimization function has the objective of optimizing the load sharing by taking into consideration the hotspots identified by the 3-stage data analytics. This function consists of four main stages: 1) The irregularity check, which is used to avoid false alarms generated by the anomaly detection function. 2) The network model, which is required to simulate the network operation in accordance with the context. The network model is used for assessing the profit and the cost of the possible optimization. 3) The decision maker which uses fuzzy rules for optimizing the cell margins following the forecasted hotspots. 4) The activation time which has an important impact on the results of the proactive optimization. The earlier the optimization is activated, the better, but it comes at the cost of an increased number of errors. To account for this, an algorithm to determine the best activation time is presented.

The authors tested their framework by simulating an urban network deployment in London. Their simulation considers the occurrence of a social event that gathers a great number of users in a cell; the number of users in the hotspot increases from an initial time t_1 to a moment of high load t_2. There is also a hotspot with a medium amount of load that represents common gathering places such as commercial areas. A Monte Carlo simulation was executed to evaluate the performance of the users with no load balancing algorithm, with a classical reactive load balancing algorithm, and with the proactive context-aware load balancing algorithm. The simulation shows that the proactive load balancing algorithm outperforms the classical alternative at maintaining the performance of the users and convergences to a solution much faster thanks to its forecasting characteristics. The simulation also shows the importance of the activation time of the optimization process for converging in a quick manner.

5.2.5 Social-Aware Optimized Fuzzy Logic Controller

In 2021, Torres et al. [14] proposed a load balancing method based on an FLC that implements classical load balancing plus a social-aware algorithm for achieving a more balanced load-sharing. This solution uses a baseline FLC that employs fuzzy rules and Boolean logic for regulating the transmission power of the cells, by applying the Power Traffic Sharing algorithm (PTS).

The FLC uses two input variables from the cellular network, the load difference between cells ($load_{Diff}$) and the transmission power deviation for each cell (ΔPtx), for computing the possible variation of the power of each cell (output value). After the first controller, a second algorithm called Social-Aware PTS (SAPTS) controller is implemented. The SAPTS algorithm uses two inputs for its operation: the first one is the radio distance between the location of a venue where a social event will take place and a close cell, whereas the second input is the azimuth of each cell located at the same site as the venue. The event location is determined by using a social-aware unit for detecting the location of user crowds in venues. The SAPTS algorithm also adjusts the rate of the power adjustment based on the radio distance; when the venue is near the cell edge, the transmission power change is slow, and when the venue is far from the cell edge, the transmission power changes are faster. In Figure 6, the architecture of the SAPTS algorithm is shown.

This load balancing method was tested in a simulation where the venue was located both in the center of a cell and at the edge of the cell. The results showed that a better sharing of UEs between cells is obtained when the social-aware algorithm is used together with the classical FLC balancing method, compared with the use of only the FLC. The proposed solution makes a differentiation between the use of machine learning methods for gathering, filtering, and analyzing the social information and the use of machine learning methods for improving the functionality of the load balancing system; in this context, the authors did not go in depth in the process used for gathering the social data, instead only mentioning that there are many online social sources from where to collect it. The authors also proposed as future work the use of more detailed mobility models for simulating the movement of the users in the area of the venue.

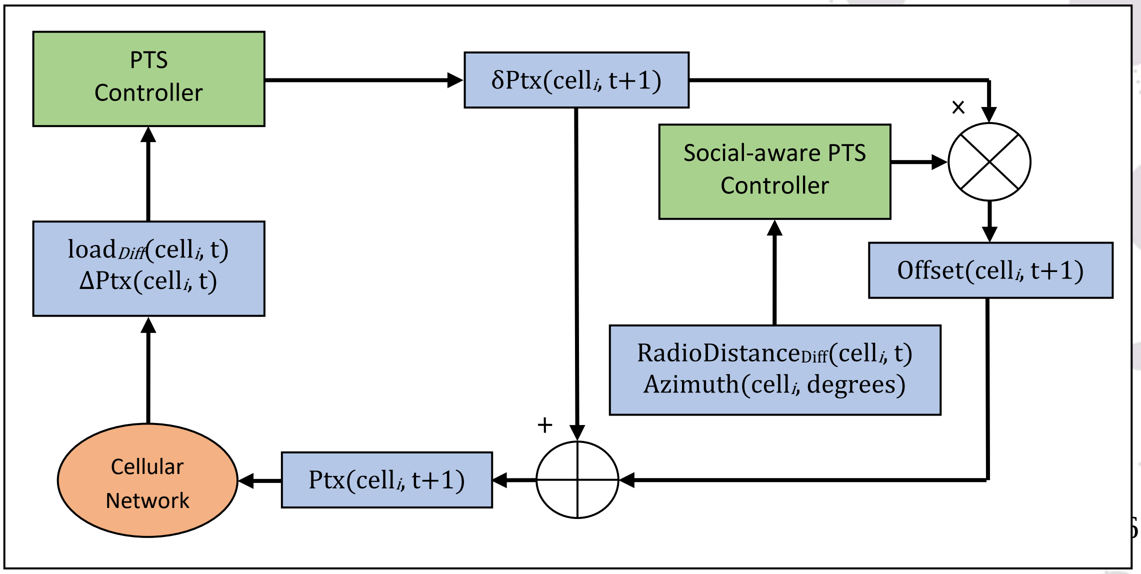

Figure 6. Architecture of the PTS and SAPTS algorithms. Taken from [14].

5.3 Characterization of a Mobile Test Environment

In this section, we will characterize a mobile test scenario over which we can evaluate the reviewed social-aware ML-based load balancing methods. This scenario is based on the guidelines of Report ITU-R M. [IMT-2020.EVAL] [24]. This report defines five different test scenarios based on the geographical environment and usage scenarios [24]. These scenarios are: 1) Indoor Hotspot-eMBB, 2) Dense Urban-eMBB, 3) Rural-eMBB, 4) Urban Macro-mMTC, and 5) Urban macro-URLLC. For this work, the Dense Urban-eMBB test environment was chosen, which is defined as "*An urban environment with high user density and traffic loads focusing on pedestrian and vehicular users*" [24]. The main parameters that this work has considered for modeling the test Dense Urban-eMMB environment are shown in Table 2.

Table 2. Parameters of the Dense Urban-eMBB test environment.

PARAMETERS	CONFIGURATION
Cellular layout	Hexagonal grid, 57 cells (3x19 sites) ISD = 0,5 Km
Total Users	1980, Venue User = 900, Random Users = 1080
Cellular area	4 Km^2 (2000 m x 2000 m)
Transmission direction	DL
Carrier frequency	4 GHz
System bandwidth	20 MHz
Frequency reuse	1
Propagation model	Shadow fading distribution: log-normal (σ_{SF} = 6)
Channel model	Multipath fading, EPA model
Mobility model	Waypoint, constant speed = 3 Km/h
Service model	Data traffic (full buffer)
Base station model	Tri-sectorized antenna, MIMO 8x8, EIRPmax=47 dBm

Scheduler	Round Robin - Best Channel
Power control	Equal transmit power per PRB
RRM features	Radio Distance (venue-cells), HO margin
Mobility pattern	Log-normal distribution

The parameters of the table will be described next for a better understanding of them:

- The cellular layout is the physical disposition of the BSs in the test scenario. In this case, 19 places are located in a hexagonal layout. In each site, there is a BS, and each BS is equipped with three radiating elements, producing 3 cells in each site. Therefore, there is a total of 57 cells in the test environment. The inter-site distance (ISD) is the distance between adjacent sites [25].
- The parameter of total users is the total number of subscribers inside the coverage area of the test scenario. In our example, there are 1980 users connected. From this, 1080 users are random and 900 users are venue attendees. In other words, these 900 users have gathered in a venue for attending a massive social event such as a sports match, a concert, or a festival.
- The cellular area is the total coverage area of the test scenario. In this case is 4 km^2.
- Transmission direction is the direction of the flow of data, whether it is from the BS to the user or from the user to the BS. In the proposed scenario, only the downlink (DL) is taken into consideration.
- Carrier frequency is the frequency used for data transmission.
- System bandwidth refers to the bandwidth of the used carrier. As the carrier is centered at 4 GHz, the system bandwidth has a range of 3.99 GHz – 4.01 GHz.
- Frequency reuse is the method for selecting and allocating frequency channel groups for all the BS within a cellular system [26]. A frequency reuse factor of 1 means that all the BSs share the same frequency channel group.
- A propagation model is a mathematical model for characterizing the propagation of radio waves through space. The propagation model used in the test environment is the log-normal shadowing model.

- A channel model is a mathematical model characterizing the effects of the communication channel trough which the wireless electromagnetic signals will propagate [27]. For the test environment, the multipath fading channel model known as Extended Pedestrian A model (EPA) is used.
- A mobility model is a mathematical model for characterizing the mobility pattern of the UEs in a network [28]. The random waypoint model is one of the most popular mobility models [29].
- Service model refers to the type of service being delivered by the network. The proposed service model focuses on packet data traffic (full buffer).
- In base station model, the main parameters of the BSs are defined. These parameters are the number of antennas of the BS, the type of antennas, and the transmission power.
- Schedulers are algorithms for distributing the capacity of the network among the existing UEs [30]. In the test scenario, the algorithms round-robin and best channel are used.
- Power control is the process of regulating the transmission power in order to achieve a good signal level. For the test scenario, the transmission power is equalized among the PRB.
- Radio Resource Management (RRM) is the management of radio transmission parameters in a wireless communication system. The most relevant features for our test environment are the radio distance between the venues and the cell and the HO margin.
- A mobility pattern describes how the users move inside a network. Real mobility patterns can only be obtained by tracking the movement of objects in the real world, unlike mobility models which are mathematical models that seek to describe these patterns [28]. The mobility pattern of the test scenario is the log-normal distribution.

6 Results

The previously described network optimization approaches aim to solve the load balancing problem in the context of SON. With the use of machine learning tools and context-aware information, they reportedly outperform classical methods of load balancing. In this work, we would like to answer the question of which one of these approaches is the most suitable to be implemented given a certain application scenario.

It is important to understand that a priori there is no better load balancing method. The choice of the algorithm depends heavily on the configuration of the cellular network and in the use case. For choosing a MLB algorithm to be used in a real-world deployment, the following criteria should be considered:

- The cellular network layout and operation parameters.
- The input parameters of the algorithm.
- The output parameters of the algorithm.
- The use case.

With these criteria, it will be decided which of the explored algorithms is the most suitable to be implemented in the test scenario.

6.1 Comparison of the Load Balancing Algorithms

Refer to Table 3 at the end of this section. There, a comparison is made between all the surveyed MLB algorithms and the test scenario using the established analysis criteria. The parameters of the system model provided by the authors do not match completely the ones of the test scenario, therefore, some of the parameters are not taken into consideration. Table 3 contains only the parameters that are mentioned by the authors in their respective works.

6.1.1 The Cellular Network Configuration

The authors of the researched methods tested their proposals in different types of system models. The task here is to asses which of these models has the closest parameters to the ones of the test environment, with the idea of evaluating which algorithm would be the easiest to be applied in the test scenario.

[18] developed their algorithms in a scenario comprised of a set of tri-sectorized macrocells placed in a hexagonal pattern and a set of picocells scattered randomly inside the macrocells. This scenario is too general as to be compared with our test scenario, however for validating their approach, the authors carried simulations in a network model consisting of one microcell, with an inter-site distance of 500 m, and three picocells randomly distributed inside the macrocell. The BSs use a frequency reuse factor of 1. This configuration is too simple in comparison with the urban eMBB test scenario and it is smaller in scale both in the number of BS and in the extension of the network. Nevertheless, when it comes to similarities, their base station model is consistent with the test scenario due to the use of a tri-sectorized antenna. They also focus on the downlink channel for their analysis and take into consideration shadowing effects just like the propagation model of our scenario.

The scenario proposed by [20] is very different from our test scenario. Their scenario is composed of a single macrocell-femtocell pair, separated by a certain distance and operating at the same frequency. This simple, tough useful model, does not take into account most of the parameters defined in our Urban Macro eMBB system, and although the authors present a framework on how to generalize their model to more complex scenarios, the results of their CAHP algorithm are not applied to this generalization. Their propagation model is also different than ours, as they use a path-loss plus fading propagation model. In addition, the mobility model is based on a straight-line trajectory that begins in the macrocell and can enter the femtocell from any direction; this is very different from the waypoint mobility model used by our test scenario. The similarities between the two models include the use of a frequency reuse factor of 1 and an analysis centered on the downlink channel with a bandwidth of 20 MHz.

[22] proposed a framework for using social-aware information in load balancing mechanisms for small cell scenarios. As a framework, their approach is given in a general

manner to be adapted to various scenarios with different types of context-aware data, load balancing mechanisms, and cellular network configurations. However, to prove the validity of their framework, the authors presented a simulation performed on a scenario composed of four small cells that serve four different areas of a shopping mall. A unique tri-sectorized macrocell is added for completing their test environment. This simple model does not fit the layout of our test scenario neither in the number of cells nor in the coverage size. Nevertheless, there are two similarities between the scenarios: the first one is the use of a frequency reuse factor of 1 and the second one is the application of the waypoint mobility model for characterizing the movement of the users inside the corridors of the shopping mall.

In the work of [3], the authors proposed a network model consisting of an urban ultra-dense small cell deployment for the city of London. This network is a heterogeneous network model, where the analysis is focused on the downlink and has a cell density of 80 cells/km^2. The channel model is a free-space channel with a bandwidth of 5 MHz and a frequency carrier of 5 GHz. The cellular area is a square with a size of 500 x 500 m. During the simulation of this network, near its center, there is a hotspot cell that will have a high load due to the occurrence of a social event. In this hotspot, the number of users grows from 18 UEs initially to 377 UEs. There is also a regular load hotspot with 78 UEs. The cellular area and the amount of the UEs considered for the simulation of this work are much lower than the amount of UEs established for the test scenario. The service model of this work is based on data traffic like in our test scenario.

[14] proposed the most similar network scenario to our test scenario. The authors tested their social-aware PTS-based algorithm in a network deployment with 5G radio parameters. The deployment consists of 19 macro-BSs that were positioned in a hexagonal pattern, with three sectors per site and separated by an inter-site distance of 500 m between them, covering a total area of 500x500 m. The layout of this network is in accordance with the layout of the test scenario except for the extension of the covered area. Another similarity between the scenarios is the radio parameters as the channel uses a frequency carrier of 4 GHz and a bandwidth of 20 MHz. The mobility model of this

work is different because the authors use the HADUMM model, although the speed of the UEs is the same as in the test scenario, 3 km/h.

Even though many parameters are considered by the test scenario that are not considered or mentioned by the reviewed works, they can be assumed as part of the analysis of the reviewed works. For example, as the goal of the MLB algorithms is to ensure a proper QoS for the users in loaded cells, the analysis of the transmission direction must focus on the downlink rather than the uplink. Using the same logic, the following parameters have been assumed for all of the works: the evaluated transmission direction is the downlink, the frequency reuse factor of the network deployment is 1 and the service model of the network is data traffic.

6.1.2 The Input Parameters of the Algorithms

The choice of a MLB algorithm is not limited only to the analysis of the configuration of the network, but also to the availability of the input parameters needed by the algorithms to operate. The reviewed works use both classical and social-aware KPIs of which, due to the nature of this work, we are more interested in the latter.

The works of [18] and [20] use the same time type of context information, the velocity of the UEs. In [20] the velocity of the users is obtained through the GPS of the UE, however, this poses a privacy problem for the users who have not consented to provide this data, making this method of acquiring the context information problematic. [18] does not mention how the velocity of the UEs is obtained, but common methods to determine this variable include Enhanced Cell Identity (E-CID), Observed Time Difference of Arrival (OTDOA), and LTE Positioning Protocol (LPP) [31].

In the CA self-optimization framework of [22], the location, activity, time, and identity are regarded as the main context data for characterizing the situation of a particular network entity, such as the users. These context data may be collected from personal devices, location systems, social networks photo and video-sharing platforms, manually introduced inputs, and various other sources. The generality of the context data proposed for the CA self-optimization framework does not allow easy comparison with the rest of the LB algorithms. However, for simulating their framework, the authors specified two sources of context data that we will use for carrying on our analysis; surveillance cameras with face-recognition technology for estimating the spatial distribution of the users and positioning systems. Similar to the collection of the velocity of the UEs by [20], both of the solutions used in this case pose a great privacy risk for the users, especially when using surveillance cameras with face-recognition technology.

The works of [3] and [14] employ public online data for the implementation of their MLB algorithms. [3] gathered 600 000 geo-tagged tweets from London and analyzed their geolocation, time, and text. On the other hand, [14] does not mention a specific type of social data for their algorithm, but it is stated that the internet provides an abundance of social data sources including social networks, calendars, open databases, browsers, event aggregators, etc. The advantage of public online data is that it is shared voluntarily by the users through their online social interactions, meaning that it is open for collection and exploitation. An example of this is when a person announces the attendance of a social event on Facebook.

6.1.3 The Output Parameters of the Algorithm

The output parameters are the network parameters tuned by the MLB algorithm for controlling the HO of a user, depending on the load of the origin and destination cells. When choosing a MLB algorithm to be implemented, it is fundamental to consider its output parameters.

The reinforcement learning algorithms of [18] learn how to adjust the CRE bias of the macro and picocells at each iteration. The CRE bias values that the macrocell can adopt are B_m = [0, 3, 6] dB and the values of the CRE bias for the picocell are B_p = [0, 3, 6, 9, 12, 15, 18] dB. The CAHP algorithm of [20] aims to find the optimal TTT value for the current context information of the network. By adjusting the TTT parameter, the network can regulate the triggering of the HO process. In the CA self-optimization framework of

[22], two MLB algorithms are discussed, the PTS and the PLS algorithms; however, the output parameter of both of these methods is the variation of the transmission power of the small cells. [3] tunes the HO margin of the BSs for offloading users from a loaded cell to an adjacent cell with free resources. Finally, in the proposal of [14], the parameter that is adjusted is the transmission power of the cell, based on the distance between the venue where a social event is happening and the cell.

6.1.4 The Use Case of the LB Algorithm

A cell is overloaded when it has to provide service to more users than its available resources; this comes with a substantial decrease in the QoS of the users. There are different instances of a cellular network where a hotspot produced by an overload can occur. For instance, during daily peak hours, the BSs can get overloaded due to the high user demand, but the cell of a network can also become overloaded during massive social events, such as sports matches or concerts, due to the excess of users gathered in a specific location.

For dealing with different instances of load balancing, the LB algorithms should be designed to be applied to different use case scenarios. This is reflected in the surveyed works where there are two different use cases in the MLB algorithms: the methods of [18] and [20] are focused on improving the MM of a cellular network while taking into consideration the load of the cells, whereas the works of [22], [3] and [14] seek to balance the load generated by the massive gathering of users in a venue due to the occurrence of a social event.

The objective of MM is to track the location of the users inside the coverage area of a wireless mobile network in order to achieve ubiquitous communication [32]. A proper MM implementation will allow a smooth HO between cells without degradation in the performance of the UEs. As discussed before, the HO process takes into account parameters such as HO hysteresis margins, TTT, and the received power of the UEs for determining the appropriate moment to change the connection of a user from one cell to another. One problem of this approach is that executing the HO process without taking into consideration the load of the target cell could reduce the performance of the UE, as the target cell may not have enough resources to serve the new UE, even though its RSRP could be higher than the one of the original cell. Taking this into consideration, it is possible to conclude that the works of [18] and [20] are not meant to solve the problem of load balancing directly, but they aim to increase the robustness of MM by taking into account the load of the BSs. The load balancing aspect appears as their algorithms prevent handovers to cells without available resources.

The methods based on public online information of [22], [1] and [14] bring proactivity to the network by being able to forecast the occurrence of social events that can create load hotspots, as the users make clear their intentions through the information they share on the internet. In this way, these methods avoid the cold start problem that appears in classical MLB algorithms and can converge to a solution more quickly [1]. The use case for which these algorithms are intended is the optimization of the network when a massive gathering of users in a venue is forecasted and the network tries to distribute the users between adjacent cells during the duration of the event.

6.2 Comparison Outcomes

The analysis showed that the work of [14] has been tested in a scenario that is the most similar to the proposed Dense urban-eMBB environment. This is clearly shown in Table 3, where the comparison between the analyzed ML algorithms is made. The work of [14] is tested in a system model that matches the Dense Urban-eMBB test scenario in the following aspects:

- Cellular layout.
- Transmission direction.
- Carrier frequency.
- System bandwidth.
- Frequency reuse.
- Propagation model.
- Service model.
- Base station model.

- Input data, and
- Use case.

Table 3. Comparison of the reviewed social-aware load balancing algorithms with the test cellular scenario.

ML Algorithms	**Authors**	**Cellular layout**	**Transmission direction**	**Carrier frequency**	**System bandwidth**	**Frequency reuse**	**Propagation model**	**Mobility model**	**Service model**	**Base station model**	**Input data**	**Output data (RRM features)**	**Use case**
		Hexagonal grid, 57 cells (3x19 sites) ISD = 0,5 Km	**DL**	**4 GHz**	**20 MHz**	**1**	**Shadow fading distribution: log-normal (σ_{SF} = 6)**	**Waypoint, constant speed = 3 Km/h**	**Data traffic**	**Tri-sectorized antenna, MIMO 8x8, EIRPmax=47 dBm**	**Social online information**	**HO hysteresis margin**	**Load balancing of a massive social event**
Reinforcement learning-based MM (MAB and satisfaction-based learning).	[18]		✓			✓	✓		✓	✓			
Markov Chain	[20]		✓		✓	✓			✓				
Context-aware enhanced fuzzy logic controller	[22]		✓			✓		✓	✓	✓			✓
BDSCAN + Fuzzy decision maker	[1]		✓			✓			✓		✓	✓	✓
PTS + SAPTS	[14]	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓

7 Discussion

As stated previously, none of the explored MLB solutions has been tested in an environment that completely matches the characteristics of the proposed test scenario. Nevertheless, this is not necessary as a real practical network deployment will never have the same conditions and assumptions as a theoretical simulation. Instead, the system model must take into consideration the most important parameters for a proper characterization of the structure of a cellular network and its operation. With this consideration, in the hereby work we have taken into account: 1) the layout and operation parameters of the cellular network, 2) the input parameters of the MLB algorithm, 3) the output parameters, and 4) its use case, to properly characterize the application scenario of the ML load balancing algorithms.

This characterization enabled a qualitative comparison between the algorithms for assessing the suitability of their application in a given cellular network deployment. The analysis showed that the algorithm of [14] is the one that possesses the most similarities with the test Dense Urban-eMBB deployment. This is because this work takes into consideration more parameters than the rest of the surveyed works for testing its algorithm, which allows it to be more easily generalized to a proposed application scenario. It has to be emphasized that this does not disprove or invalidate the findings of the rest of the works, but that their proposals have to be tested in a more detailed environment to improve the confidence that their results can hold in a real-life setting.

8 Conclusions

- A qualitative comparison between various machine learning-based social-aware load balancing methods was performed by establishing a set of criteria for characterizing their application scenarios.
- A systematic literature review was done to assess the state of the art about the use of ML and social-aware information for solving the load balancing problem in the context of SON. The literature review was conducted using the guidelines given by [17]. The systematic literature review started by properly defining the research problem through the proposition of research questions. Next, a set of inclusion and exclusion criteria for conducting the review was established and finally, the review was conducted taking into consideration the aforementioned criteria. Through this review, five MLB methods were selected for analysis.
- A set of four parameters was proposed as criteria for characterizing the application scenario of the MLB algorithms, to allow a comparison between them. These criteria are 1) the cellular network layout and operation parameters, 2) the input parameters of the algorithm, 3) the output parameters of the algorithm, and 4) its use case.
- The social-aware MLB solution of [14], based on the PTS and SAPTS algorithms, was chosen as the most suitable to be applied in a Dense Urban-eMBB test scenario.

9 Recommendations

- The execution of a systematic literature review requires defining clear limits in the scope of the search for obtaining meaningful results. The search scope must be limited in time, specifying how old the works that will be reviewed may be. It also has to be limited in the scope of topics that must be reviewed; for this, it is a good practice to define a set of keywords to look for when doing a literature review and a set of keywords to avoid, as they might be related to the research topic but they are not the focus of the study. For performing a systematic literature review, it is recommended to follow the method of [17].
- Modeling a cellular network environment can be challenging as it contains a great number of variables for defining its structure and operation. The parameters of the network are also dependent on the environment where it will be deployed, whether it is an urban or rural setting. The main parameters of current NGMN such as LTE or New Radio are defined in the technical reports of the main organizations working for the standardization of this technology such as 3GPP, ETSI, and the ITU. These reports can be used as a starting point for easing the process of modeling a NGMN.

10 Bibliography

[1] B. Ma, G. Weisi and J. Zhang, "A Survey of Online Data-Driven Proactive 5G Network Optimisation Using Machine Learning," *IEEE Access*, pp. 35606-35637, 2020.

[2] P. Muñoz, *Optimization of Mobility Parameters using Fuzzy Logic and Reinforcement Learning in Self-Organizing Networks*, Málaga: Universidad de Málaga, 2014.

[3] B. Ma, B. Yang, Y. Zhu and J. Zhang, "Context-Aware Proactive 5G Load Balancing and Optimization for Urban Areas," *IEEE Access*, vol. 8, pp. 8405-8417, 2020.

[4] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," in *Handheld and Ubiquitous Computing*, Karlsruhe, 1999.

[5] P. V. Klaine, M. A. Imran, O. Onireti and R. D. Souza, "A Survey of Machine Learning Techniques Applied to Self Organizing Cellular Networks," *IEEE Communications Surveys & Tutorials*, pp. 2392-2431, 2017.

[6] J. Moysen and L. Giupponi, "From 4G to 5G: Self-organized network management meets machine learning," *Computer Communications 129*, pp. 248-268, 2018.

[7] H. Fourati, R. Maaloul, L. Chaari and M. Jmaiel, "Comprehensive survey on self-organizing cellular network approaches applied to 5G networks," *Computer Networks 199*, p. 108435, 2021.

[8] E. Gures, I. Shayea, M. Ergen, M. H. Azmi and A. El-Saleh, "Machine Learning-Based Load Balancing Algorithms in Future Heterogeneous Networks: A Survey," *IEEE Access*, pp. 37689-37717, 2022.

[9] Y. Xu, W. Xu, Z. Wang, J. Lin and S. Cui, "Load Balancing for Ultradense Networks: A Deep Reinforcement Learning-Based Approach," *IEEE Internet of Things Journal*, pp. 9399-9412, 2019.

[10] A. Mohajer, M. Bavaghar and H. Farrokhi, "Mobility-aware load Balancing for Reliable Self-Organization Networks: Multi-agent Deep Reinforcement Learning," *Reliability Engineering & System Safety,* vol. 202, p. 107056, 2020.

[11] O. Sallent, J. Pérez-Romero, J. Sánchez-González, R. Agustí, M. A. Díaz-Guerra, D. Henche and D. Paul, "A roadmap from UMTS optimization to LTE self-optimization," *IEEE Communications Magazine,* vol. 49, no. 6, pp. 172-182, 2011.

[12] 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 Transactions on Vehicular Technology,* vol. 62, no. 5, pp. 1962-1973, 2013.

[13] P. Muñoz, R. Barco and I. de la Bandera, "Optimization of load balancing using fuzzy Q-Learning for next generation wireless networks," *Expert Systems with Applications,* vol. 40, no. 4, pp. 984-994, 2013.

[14] R. Torres, S. Fortes, E. Baena and R. Barco, "Social-Aware Load Balancing System for Crowds in Cellular Networks," *IEEE Access,* vol. 9, pp. 107812-107823, 2021.

[15] R. Lee and K. Sumiya, "Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection," in *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, New York, 2010.

[16] K. Nguyen Doan, T. Van Nguyen, T. Q. S. Queck and H. Shin, "Content-Aware Proactive Caching for Backhaul Offloading in Cellular Network," *IEEE Transactions on Wireless Communications,* vol. 17, no. 5, p. 3128–3140, 2018.

[17] P. V. Torres-Carrión, C. S. González-González, S. Aciar and G. Rodríguez-Morales, "Methodology for systematic literature review applied to engineering and

education," in *2018 IEEE Global Engineering Education Conference (EDUCON)*, 2018.

[18] M. Simsek, M. Bennis and İ. Güvenç, "Context-aware mobility management in HetNets: A reinforcement learning approach," in *IEEE Conference on Wireless Communications and Networking*, 2015.

[19] S. Ross and B. Chaib-draa, "Satisfaction Equilibrium: Achieving Cooperationin Incomplete Information Games," in *Conference of the Canadian Society for Computational Studies of Intelligenc*, Québec, 2006.

[20] F. Guidolin, I. Pappalardo, A. Zanella and M. Zorzi, "Context-Aware Handover Policies in HetNets," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 1895-1906, 2016.

[21] 3GPP, *3GPP TS 36.331: Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification*, 2011.

[22] A. Aguilar-Garcia, S. Fortes, A. Fernandez Duran and R. Barco, "Context-Aware Self-Optimization: Evolution Based on the Use Case of Load Balancing in Small-Cell Networks," *IEEE Vehicular Technology Magazine*, vol. 11, no. 1, pp. 86-95, 2016.

[23] J. Ruiz-Avilés, S. Luna-Ramírez, M. Toril, F. Ruiz, I. de la Bandera, P. Muñoz, R. Barco, P. Lázaro and V. Buenestado, "Design of a computationally efficient dynamic system-level simulator for enterprise LTE femtocell scenarios," *Journal of Electrical and Computer Engineering*, vol. 2012, 2012.

[24] WP5D, "Draft new Report ITU-R M.[IMT-2020.EVAL] - Guidelines for evaluation of radio interface technologies for IMT-2020," ITU, 2017.

[25] MathWorks, "SINR Map for a 5G Urban Macro-Cell Test Environment," [Online]. Available: https://la.mathworks.com/help/phased/ug/sinr-map-for-a-5g-urban-micro-cell-test-environment.html. [Accessed 5 August 2023].

[26] T. S. Rappaport, Wireless Communications--Principles and Practice, 2nd ed., Prentice Hall, 2002, p. 26.

[27] MathWorks, "Channel Model," MathWorks, [Online]. Available: https://www.mathworks.com/discovery/channel-model.html. [Accessed 7 August 2023].

[28] C. Schindelhauer, "Mobility in Wireless Networks," in *International Conference on Current Trends in Theory and Practice of Computer Science*, 2006.

[29] S. Mao, "Chapter 8 - Fundamentals of communication networks," in *Cognitive Radio Communications and Networks*, Academic Press, 2010, pp. 201-234.

[30] J. Ramis, L. Carrasco, G. Femenias and F. Riera-Palou, "Scheduling algorithms for 4G wireless networks," in *Personal Wireless Communications*, Prague, 2007.

[31] R. Saraiva Campos, "Evolution of Positioning Techniques in Cellular Networks, from 2G to 4G," *Wireless Communications and Mobile Computing*, 2017.

[32] Y. Fang and W. Ma, "Mobility Management for Wireless Networks: Modeling and Analysis," in *Wireless Communications Systems and Networks*, 2004, p. 473–512.