IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 5, 2022, accepted March 16, 2022, date of publication March 23, 2022, date of current version April 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3161511

Machine Learning-Based Load Balancing Algorithms in Future Heterogeneous Networks: A Survey

EMRE GURES[1], IBRAHEEM SHAYEA[1], MUSTAFA ERGEN[1], MARWAN HADRI AZMI[2], AND AYMAN A. EL-SALEH[3], (Senior Member, IEEE)

[1]Department of Electronics and Communication Engineering, Faculty of Electrical and Electronics Engineering, Istanbul Technical University (ITU), 34467 Istanbul, Turkey
[2]Wireless Communication Centre, Faculty of Engineering, School of Electrical Engineering, Universiti Teknologi Malaysia (UTM), Johor Bahru 81310, Malaysia
[3]Department of Electronics and Communication Engineering, College of Engineering, A'Sharqiyah University (ASU), Ibra 400, Oman

Corresponding author: Emre Gures (gurese18@itu.edu.tr)

This work was supported by the 2232 International Fellowship for Outstanding Researchers Program of TÜB TAK conducted at Istanbul Technical University (ITU) under Project 118C276; in part by Universiti Sains Islam Malaysia (USIM), Malaysia; in part by the Ministry of Higher Education Malaysia (MOHE) under the Fundamental Research Grant Scheme under Grant FRGS/1/2019/TK04/UTM/02/34; in part by Universiti Teknologi Malaysia (UTM) through the Collaborative Research Grant (CRG) under Grant R.J130000.7351.4B468; and in part by the Higher Institution Centre of Excellence (HICOE) under Grant R.J130000.7851.4J413 and Grant R.J130000.7851.4J493.

ABSTRACT The massive growth of mobile users and the essential need for high communication service quality necessitate the deployment of ultra-dense heterogeneous networks (HetNets) consisting of macro, micro, pico and femto cells. Each cell type provides different cell coverage and distinct system capacity in HetNets. This leads to the pressing need to balance loads between cells, especially with the random distribution of users in numerous mobility directions. This paper provides a survey on the intelligent load balancing models that have been developed in HetNets, including those based on the machine learning (ML) technology. The survey provides a guideline and a roadmap for developing cost-effective, flexible and intelligent load balancing models in future HetNets. An overview of the generic problem of load balancing is also presented. The concept of load balancing is first introduced, and its purpose, functionality and evaluation criteria are then explained. Besides, a basic load balancing model and its operational procedure are described. A comprehensive literature review is then conducted, including techniques and solutions of addressing the load balancing problem. The key performance indicators (KPIs) used in the evaluation of load balancing models in HetNets are presented, along with the concurrent optimisation of coverage (CCO) and mobility robustness optimisation (MRO) relationship of load balancing. A comprehensive literature review of ML-driven load balancing solutions is specifically accomplished to show the historical development of load balancing models. Finally, the current challenges in implementing these models are explained as well as the future operational aspects of load balancing.

INDEX TERMS Mobility management, load balancing, heterogeneous networks, handover, handover problems, handover self-optimization, mobility challenges, machine learning, 5G network, 6G network, future ultra-dense.

I. INTRODUCTION

The enormous increase in the use of smart devices and applications, which form information and communication technology, has significantly raised the demand for mobile broadband services with higher data rates and improved quality of service (QoS) [1]. To meet such an enormous service demand, the next-generation sixth generation (6G) of wireless networks must establish advanced broadband, massive access, ultra-reliable and low-latency service capabilities that are stronger and smarter than what is offered in the fifth generation (5G) wireless network. The 6G network structure is envisioned to be extremely heterogeneous, heavily deployed and highly dynamic [2]. The benefits of ultra-wideband, ultra-wide access, ultra-reliability and low latency to be achieved in 6G technology will also bring about crucial

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa Zaman Chowdhury.

questions such as how such networks would be managed and controlled.

One of the main challenges in 6G mobility management is ensuring a fair load distribution among cells in HetNets [3], [4]. Load balancing is an important consideration for managing available radio resources within the network. Small cells are placed in high-traffic areas to increase network capacity. They provide additional resources to macro cells in certain locations, fill in the "coverage gaps" of the macro cell layer and reduce the load levels of dense cells. Since the various types of cells composing the HetNet have different transmit powers, using the traditional user association scheme based on the maximum received power causes a heavy load imbalance in the network [5], [6]. In most cases, users are associated with macro cells that provide the strongest downlink signal. This load imbalance causes low-powered small cells to waste remaining resources due to the small number of associated users, while users associated with heavily loaded base stations (BSs) compete for insufficient network resources. Effective load balancing approaches are required to avoid overloading cells and the depletion of radio resources. Figure 1 shows the working principle of an exemplary load balancing model in a HetNet. Here, users in highly loaded macro cells are handed over to less-dense macro cells and small cells to provide load balancing in the network.

The capacity to handle large volumes of data and provide high data rate connectivity per device is a key requirement for 5G/6G intelligent networks [7]. Load balancing is a promising solution to efficiently handle higher data rates, improve system capacity by regulating cell congestion and manage wireless resource allocation across multiple connections. It also offers enhanced system performance, higher resource utilisation and lower operating costs, thus increasing the adaptability and availability of the network.

Mobile networks have already created a large data source to efficiently manage HetNets by devising more informed decisions. In this context, artificial intelligence (AI)/ML is a great opportunity since it can provide meaningful insights using the data analysis currently available. The promising aspect of AI/ML approaches is their ability to automatically learn system experience, predict future scenarios and adapt to operating environments [8]. With AI/ML, users can select cells that will optimally serve them, dynamically manage interconnection with multiple cells and select the most suitable HO target cells to ensure service continuity [7]. With AI/ML, BSs can optimise system parameters (such as mobility parameters) to provide load balancing and enhance network strength. Applying AI/ML tools will enable valuable insights by training the observed data. Different functions can be learned to support forecasting, decision making and optimisation for balancing loads in 6G HetNets.

Table 1 provides an overview of survey papers available in the literature on load balancing. The table presents extensive information of existing studies. The characteristics of the presented survey have been specified and compared with other studies. This paper provides readers a useful guide towards creating cost-effective and flexible AI/ML-based models to solve the load balancing problem in HetNets. In this paper, a comprehensive review of the load balancing problem is first presented. For this purpose, the concepts, objectives, functionality and evaluation criteria of load balancing are explained. A basic load balancing model is then introduced, with each implementation step explained. An extensive explanation is provided from the literature regarding managing the load balancing problem as well as the techniques used to solve this issue. The KPIs employed in the load balancing models are further introduced, together with their formulation. The relevance of load balancing to concurrent optimisation of coverage (CCO) and mobility robustness optimisation (MRO) is extensively explained as well. The control strategies of load balancing techniques are also mentioned. A comprehensive summary is provided to highlight technical details such as the application of ML-oriented load balancing models, the steps taken in their implementation, their performance analysis and their shortcomings. The challenges in implementing load balancing models and future working directions are also mentioned.

This paper is structured as follows: Section II provides an overview of the load balancing problem. Section III provides an overview of ML and a comprehensive review of ML-based load balancing models available in the literature. Sections IV and V provide a brief introduction and summary of the challenges in load balancing problems and future research directions, respectively. Section VI concludes the paper.

II. LOAD BALANCING OVERVIEW

Load imbalance is an inherent problem of future HetNets. The main reasons include random user access to the network, the changes that occur over time and the varying business requirements. This section provides an overview of the load balancing problem.

A. CONCEPT OF LOAD BALANCING

Load balancing is the even distribution of cell loads among adjacent cells, or the transfer of traffic from congested cells to more available cells so that the use of radio resources remains highly optimised. All users associated with a cell share the bandwidth of that cell. A loaded condition occurs when the cell's workload exceeds or approaches capacity due to the maximum number of users per cell reaching the limit. It also occurs when admission control begins to block new users from entering the network, trying to prevent throughput from increasing. Ideally, the scheduler should be able to allocate the necessary physical resource blocks (PRBs) to users for each service that requires a certain QoS. As more and more users join the network and consume PRB resources, the load balancing mechanism should kick strat at some point and begin to proactively redirect users to other cells to avoid overloading or congestion. A load balancing function/algorithm is used to prevent cell overload and consequent performance degradation.

TABLE 1. Comparison of existing surveys in the literature on load balancing.

Survey	Year	Description
[9]	2014	• Mythical assumptions in cellular networks are re-discussed in the context of a load-balanced HetNet to dispel these myths. • Approaches to load balancing in HetNets are explored and compared. • Several open areas for future exploration of load balancing are identified.
[10]	2014	• Emphasis is placed on appropriate techniques for self-optimisation in cellular networks, highlighting current load balancing techniques.
[11]	2016	• A review of adaptive cell selection techniques is presented to achieve improved load balancing.
[12]	2016	• A systematic review of load balancing algorithms in the cloud environment is accomplished. In this direction, the studied algorithms are compared, and their application in cloud computing systems is examined. The complexities of these algorithms are also addressed.
[13]	2017	• The existing load balancing mechanisms in cloud computing are analysed. • Load balancing mechanisms are classified and the pros and cons of each class are evaluated. • New research topics on load balancing algorithms in cloud computing are outlined.
[14]	2018	• The load balancing mechanisms used in software-defined networking (SDN) are divided into two main categories (deterministic and non-deterministic approaches) and the pros and cons of selected load balancing algorithms are analysed. The KPIs applied in the algorithms are also explored. • Key research directions are identified for future work by examining the challenges of selected algorithms.
[15]	2019	• Load balancing methods in Internet of Things (IoT) are classified into two basic classes (centralised and distributed methods). A detailed evaluation of the studied methods, with their merits and shortcomings, is also accomplished. • Load balancing challenges in IoT systems, evaluation parameters and critical areas for future research are highlighted.
[16]	2019	• The fog computing architecture is defined and the load balancing algorithms applied in fog computing are examined. • An analysis based on different measurements of fog calculation and simulation tool is offered.
[17]	2019	• The load balancing techniques used in cloud computing are classified, and the advantages, disadvantages, concepts and challenges of each class are discussed.
[18]	2019	• Information on the tasks performed in data centres and the cloud environment are provided, as well as the load balancing techniques used.
[19]	2020	• A systematic literature review of load balancing techniques in SDN is provided. • Current load balancing techniques in SDN are classified as either AI-based or traditional, with focus on improvements and limitations. • KPIs used in current load balancing techniques are introduced. • Recommendations for future research are discussed.
[20]	2020	• SDN and OpenFlow technologies are outlined along with their impact on load balancing. • An overview of load balancing schemes on SDN is provided, with the highlight on research challenges, current solutions and future research directions. • The KPIs used in load balancing models are summarised.
[21]	2020	• A thematic classification of load balancing in SDN is presented, taking into account issues such as the objectives of load balancing, the SDN architecture associated with load balancing, data/control plane load balancing techniques and the KPIs used for such techniques. • The current challenges and future research topics with load balancing techniques using SDN are highlighted.
[22]	2021	• Examines the background of the load imbalance problem in HetNets. It details the importance of load balancing and the advantages of including SDN in a load balancing strategy. • Analyzes SDN-based HetNet load balancing strategies according to the research area from two aspects: data transmission -based load balancing strategy and resource-based load balancing strategy. • Summarizes the problems of existing solutions through the analysis and predicts and analyzes the future development direction of SDN-based load balancing strategies.
[23]	2021	• Analyze the concept and development of SDN. It also provides a systematic review of the load balancing approach that can be implemented in SDN.
[24]	2021	• Presents a systematic review of current load balancing methods, considering the importance of fault tolerance in load balancing algorithms used in cloud computing. • Classifies existing load balancing techniques into two main categories as centralized and distributed, and examines qualitative parameters such as scalability, response time, reliability, availability, efficiency and overhead.
[25]	2021	• Examines the current load balancing algorithms in the literature used in the cloud environment and presented the problems, weaknesses and strengths of the algorithms and a compilation of flowcharts. • A fault-tolerant model is proposed to address the fault tolerance problem in load balancing algorithms. It emphasizes research directions for future work through this model.
[26]	2021	• Provides a comprehensive review of load balancing models in the cloud environment. It explains the load balancing algorithms used in this field and then compared performance between them.
This survey		• Provides a comprehensive review of ML-based load balancing models in future HetNets. Makes a detailed analysis of load balancing, including concept, algorithms, addressing, KPI, problems and their control. Explains the working principles of SON functions such as CCO and MRO, and provides detailed information about their relationship to load balancing and their joint optimization. • Explains the steps taken in the implementation of ML-based load balancing models in the literature from 2013 to 2021, performance analysis and the shortcomings of the models. By presenting them with a historical flow, the development of these models, how solutions were brought to the deficiencies in the models and how these solutions led to results were conveyed to the readers. • Challenges in existing solutions are identified, and new research directions that can alleviate these challenges and help create a more stable network structure are explored and discussed.

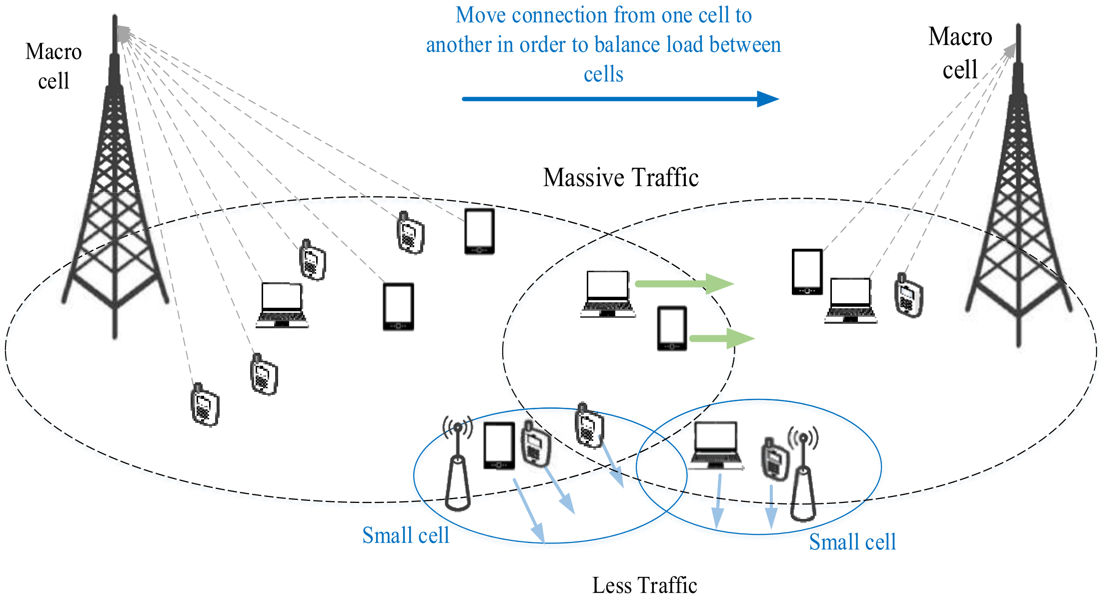
Macro cell
Move connection from one cell to another in order to balance load between cells
Macro cell
Massive Traffic
Small cell
Small cell
Less Traffic

FIGURE 1. Balancing loads between cells based on system capacity availability: users at the cell edge of macro cells with massive traffic will be handed over to small cells with low traffic and to also macro cells with relatively less loaded [4].

Load balancing is a vital function in HetNets or multi-layer network deployments. HetNets have been primarily adopted to improve capacity and coverage in areas with unequal user distribution. Small cells are usually placed to provide extra capacity in locations with high user demand, while macro cells are used to provide coverage in remaining areas. Macro cells have higher transmission power than small cells. Users prefer the cell with the strongest received signal strength (RSS) when associating with a cell. Since the number of user clusters connected to small cells will be much less than the macro cell, the available resources of small cells will not be fully utilised and the competition for available resources in the macro cell will remain high. Load imbalance between cell types leads to unfair data rate distribution among users within the network, as well as inconsistencies in their quality of experience (QoE) [27]. Load balancing is basically responsible for detecting the source of load imbalance in the network and fairly reassigning users to all available cells in a region [28]. This technique ensures that the radio resources of the network are efficiently used, while providing service quality to users.

1) OBJECTIVE

The aim of load balancing is to achieve the following:

- Optimise cell re-selection/HO parameters to minimise the number of HOs and redirections required to maintain load balance between neighbouring cells,
- Improve system capacity by regulating cell congestion,
- Establish efficient and effective management for optimum performance.

2) FUNCTIONALITY

The load balancing algorithm distributes the UEs camped in or connected to a cell to balance the traffic load. This function can be performed by delaying or advancing the HO of UEs between cells. Load balancing includes the following processes:

- The load report function measures the load for each cell covered by its own BS, and this measurement information is driven by exchanging cell-specific load information between neighbouring BSs via the interface X2 or S1.
- Based on this information, an algorithm checks if it is necessary to redistribute the load among neighbouring cells. If a change is needed, the source cell sends a mobility change request to the neighbouring cell.
- An algorithm predicts whether the HO parameter settings must be changed. If needed, communication occurs between the cells involved to suggest changes in the neighbouring HO trigger settings for the neighbouring cell.

3) EVALUATION CRITERIA AND EXPECTED RESULTS

The expected results are as follows:

- According to the HO mechanism, some UEs at the cell boundary are handed over to less loaded cells.
- In the new state, the cell load is balanced.
- System capacity is increased.

B. PROCEDURE OF LOAD BALANCING ALGORITHMS

In Figure 2, the procedure of a load balancing algorithm is explained to provide an insight into the working principle. A load balancing algorithm decides to distribute users that are either camped in or connected to a cell to balance the traffic load. This can be achieved by delaying or advancing HO of users between cells. The load balancing function collects load performance metrics (e.g. radio resource usage, load indicators, etc.) or notifications (e.g. threshold crossing of certain metrics (Section II-D)). In order to decide on the most suitable candidate cell for load balancing, in addition to the load of the service BS, the loads of neighboring cells must also be known. The load balancing function analyzes the load measurements to determine if the configuration of the HO and/or re-selection parameters is required to optimize the traffic load distributions. Performance measurements (e.g. HO failure (HOF), call drops, throughput, etc.) are performed to evaluate the performance of the load balancing optimization, and the HO and/or re-selection parameters can be updated based on the metrics. The basic load balancing algorithm presented in Figure 2 is explained in the following points from 1 to 6. The choices (a) and (b) in the fifth step represent the situations whether the condition is met or not, respectively. The operating procedure of the load balancing algorithm can be logically interpreted as follows:

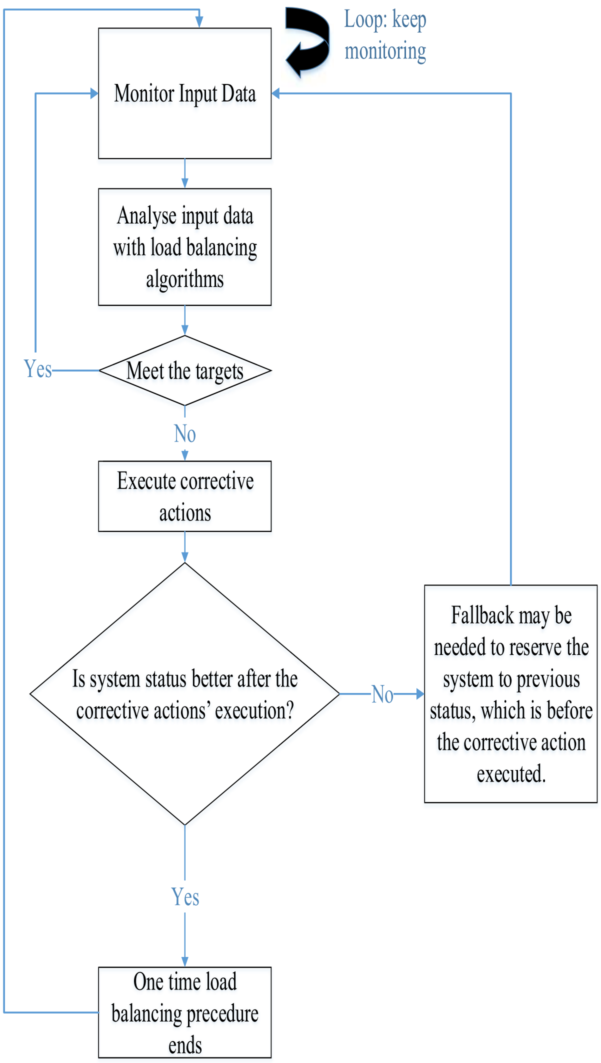
Loop: keep monitoring
Monitor Input Data
Analyse input data with load balancing algorithms
Yes
Meet the targets
No
Execute corrective actions
Is system status better after the corrective actions' execution?
No
Fallback may be needed to reserve the system to previous status, which is before the corrective action executed.
Yes
One time load balancing precedure ends

FIGURE 2. Flowchart of load balancing procedure.

1. The load balancing function continues to monitor load levels in cells to detect input data load imbalance.
2. The load levels of cells are calculated, analysing whether the load levels show a balanced distribution.
3. If a load imbalance is detected, the optimisation algorithm is triggered.
4. According to the load level information of the cells and the load balancing policy, the most suitable target cell is determined to offload the traffic. The target cell is determined by criteria such as cell load levels, RSRP, cell types (e.g. mmWave cells can be prioritized with high channel bandwidth). Necessary data transfer and metadata changes are made in order for the relevant users to be HO to the determined target cells. Corrective actions refer to the totality of HO events performed for load balancing purposes.
5. The load balancing function evaluates the result of the executed operations.
    a) If the network condition is satisfactory after taking action to achieve load balancing, the load balancing process ends once.
    b) If the network state is unsatisfactory after taking action to achieve load balancing, a fallback may be required to revert to the network configuration before the actions are taken.
6. Load balancing functionality returns to monitoring the input data.

C. HOW LOAD BALANCING CAN BE ADDRESSED?

Numerous efforts have been made to achieve load balancing in HetNets. The load balancing procedures have been grouped under the following headings.

1) CELL RANGE EXPANSION (CRE)

CRE, the standardised technique of the third generation partnership project (3GPP), is a promising method for balancing loads in HetNets. Basically, cell coverage is expanded or narrowed by adding a bias to the pilot power value of the cell. New alternatives are thus created which users can associate with. In user association, small cells with biased power may become more attractive to some users than macro cells. Thus, by offloading user traffic from macro cells to small cells, the total system serves more traffic and users achieve higher throughput [29].

The main goal of CRE is to offload traffic from over-utilised cells to under-utilised cells so that the total network load can be shared more evenly between cells. Reshaping cell coverage areas is accomplished by adjusting mobility parameters, similar to the methods used in cell re-selection or the HO algorithm [30]. One possible method to accomplish this is to set the cell individual offset (CIO) values of the cells. At a specific t time, a UE's HO from the serving cell i to the neighbouring target cell j is triggered according to the "A3" condition [31].

$$RSRP_j + CIO_{j\to i} = RSRP_i + CIO_{i\to j} + Hys; \quad (1)$$

$RSRP_i$ and $RSRP_j$ are the measured values of the reference signal received power (RSRP) from cells i and j, respectively. $CIO_{i\to j}$ is the CIO of cell i with respect to cell i, $CIO_{j\to i}$ is the CIO of cell j with respect to cell i and Hys is the HO hysteresis, which is usually a fixed value to avoid frequent HOs. $CIO_{i\to j}$ can be interpreted as the offset value indicating stronger (or weaker) measured RSRP of cell i compared to the measured RSRP of cell j. Similarly, $CIO_{j\to i}$ is the added value that makes the measured RSRP of cell j appear stronger (or weaker) compared to the measured RSRP of cell i. Generally, $CIO_{i\to j}$ is chosen symmetrically ($CIO_{i\to j} = CIO_{j\to i}$) to avoid the ping-pong effect, i.e., to prevent the user handed over from one cell to the other from bouncing directly back. Reducing the CIO value can trigger the HO of the user from cell i to cell j, thus offloading the load from cell i to cell j. Of course the opposite can also occur when the CIO value increases. In this case, a good strategy to achieve load balancing would be to assign high CIO values to under-utilised cells and low CIOs to over-utilised cells. The key aim of the CRE technique is to determine the best CIO tuning policy which optimally triggers user HOs to achieve network load balance.

Several problems do occur if the CIO value is not set properly. If the CIO is too low, a large number of users cannot associate with these cells due to the narrow coverage of small cells. The resources of these cells (e.g., frequency band/power) will not be entirely utilised, resulting in poor system performance. If the CIO value is too high, the small cell's coverage will expand more than necessary, associating more users than it can serve. Most users associated with small cells cannot be programmed since they are further from small BSs and have faulty RSRP which can cause scheduling outage [32]. Users that are relatively close to the macro BS will also experience dramatic interference.

Most previous studies on CRE have recommended the use of fixed CIO values. Since small BSs are deployed at spatially diverse and ever-changing traffic densities, coverage must respond quickly to traffic density changes. Fixed CIO usage is effective at offloading traffic, but this can lead to unfair load sharing due to varied user densities. On the other hand, an adaptive CIO determination strategy assigns cell-specific CIO values to small BSs according to the load condition, resulting in a fair load distribution. In [33], an adaptive load balancing algorithm was proposed for a homogeneous small cell network. Overloaded cells are detected using an adaptive threshold value. It predicts the load status of serving and adjacent neighbouring cells and adjusts the CIO values of serving and target cells while considering cell pair conditions. However, this algorithm cannot provide network-wide load balancing as it only considers the adjacent cells of overloaded cells. In [34], a cell selection scheme was presented where the CIO value is adaptively adjusted based on the performance of the signal to interference plus noise ratio (SINR). By summing the SINRs of macro cell users, the cumulative distribution function graph is drawn and a SINR threshold value is determined accordingly. The algorithm determines the CIO value of each user based on the SINR threshold. However, feedback from each UE causes latency in the system where the load state of the cell cannot be efficiently predicted. In [35], a coordinated CRE scheme is proposed to analytically calculate the joint CIO value in small cells and macro cells. However, HOs among BSs in dense HetNets have not been properly studied. The recent complexity of having different radio systems in HetNet makes it difficult to set CIO parameters in this way. Due to the dynamic characteristics of the network environment, such a complex problem requires solutions that can efficiently adapt to changes. Models in the literature have been used to appropriately adjust CIO values for better redistribution of traffic between cells. Greater QoS can be achieved as well as adequate network capacity with minimal human intervention in network management through the integration of ML. These models are extensively examined in Section III.

Users in the extended region associated with small cells by CRE technique may be affected by high co-channel intercell interference in the downlink from macro cells if the operating frequency of the macro cell and the small cell is the same. As seen in Figure 3, although the user in the second region is associated with the small cell, it is exposed to high interference due to the high RSRP of the macro cell. Data channel interference for extended coverage is mitigated by the interference cancellation in the UE or by coordinated resource allocation, such as the application of dynamic and self-organised interference mitigation techniques [36]. The enhanced inter-cell interference coordination (e-ICIC) is a mechanism that alleviates the serious interference problem caused by macro cells for cell edge users. The e-ICIC technique allows small cells and macro cells to use radio resources at different time intervals (subframes) to prevent inter-cell

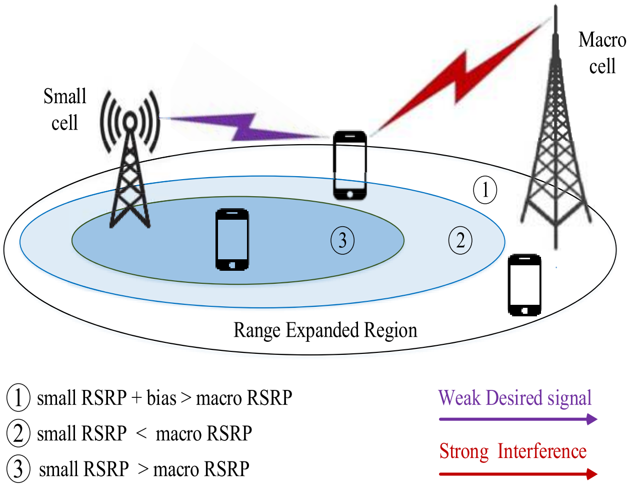
Macro cell
Small cell
1
2
3
Range Expanded Region
1 small RSRP + bias > macro RSRP
2 small RSRP < macro RSRP
3 small RSRP > macro RSRP
Weak Desired signal
Strong Interference

FIGURE 3. Users in the extended area may experience high co-channel interference due to the high RSRP of the macro cell.

interference. Users in extended range areas are programmed in almost blank subframes (ABS) to protect them from the exposure of strong cross-layer interference. The residual normal subframes are allocated to users near small cells and macro cells. Users with poor channel conditions can thereby increase their SINR as they avoid the interference effect caused by macro cells in data symbols [37]. The ABS ratio is an important criterion that can be used during the cell selection procedure. It is expressed as the ratio of ABS to non-ABS. A low ABS ratio causes offloaded users to have low average data rates due to overloaded users in the ABS [38]. With the proper configuration of the CIO value and ABS ratio, the QoE of cell edge users and the overall system performance can be enhanced [39].

2) CELL BREATHING

Cell breathing is a load balancing technique that controls the transmission power of beacon signals of access points to adjust cell coverage areas. Load balancing is accomplished using power control algorithms that reduce (or increase) the power level to a narrow (or wide) scope of over-loaded or under-loaded cells. An overloaded access point reduces the transmit power of the beacon signal, thus lessening the likelihood of new users discovering themselves. It further allows some users to connect to neighbouring access points or connect to the cellular network if another access point is not within range. It should be noted that the transmission power of data packets does not decrease. This technique does not affect the loss rate or sending rate of data packets, it only affects the access point/user associations [40].

Although the cell breathing technique is widely employed in WLAN systems, it also has a few applications in LTE systems. The main reasons are: it automatically adjusts the size of the cell, increasing the probability of coverage holes, and it implements power allocation where the LTE downlink cannot change the reference signal strength without adjusting the data power [41]. In [42], a load balancing model based on a hybrid LTE/WLAN cell breathing technique was proposed. The model allows the technique to simply associate with existing WLAN networks, similar to traditional WLAN priority network association, reducing complex coordination and additional signal overheads. In [40], various load balancing algorithms based on cell breathing were proposed for Wi-Fi/cellular HetNets. The algorithms define the load value thresholds of access points. When a cell's load exceeds the threshold, the offload amount from the cellular network is limited via coverage holes.

3) DATA ANALYTICS

Lately, data analytics-based load balancing algorithms that predict traffic in hotspots have been gaining popularity. Load balancing models, in which data network analytics are integrated, basically consist of four stages: collecting data, filtering the collected data, analyzing the data and optimizing the network through the analyzed data. There are different ways of collecting raw data. The first is to create a dataset by collecting information from volunteers. This method can provide a complete dataset, but the data collection process is time consuming. [43]–[45] use data from the Qiangsheng Taxi Company in Shanghai to examine real-life GPS-based vehicle mobility tracks. The dataset was created from data collected from approximately 25% of all taxis in Shanghai. Each taxi equipped with GPS devices periodically sends reports to the data collector. The dataset contains information such as taxi ID, operating status, timestamp, orientation, vehicle movement speed, latitude, longitude. In [46], data was collected from a group of students at the University of Bologna for fifteen months via a smartphone detection platform to create the dataset. The second method is to create a dataset by scanning through social media applications or application programming interface [47]. This method is economical and fast, but it is difficult to create a complete dataset. The last method is to create a dataset through service providers. After obtaining the raw data, the features to be analyzed are determined and irrelevant data in the dataset is filtered, thus reducing the size of the dataset and providing a numerical expression of the features to be analyzed (for instance, the coordinates of the users, the intensity of use of the social platform from which the data is provided in a specific region). After this stage, the data is ready for data analytics. The spatio-temporal traffic pattern is created by ML methods with data analytics functionality so that hotspot changes can be analyzed. Finally, the overload in hotspots determined by load balancing algorithms is transferred to low-density cells, optimizing the overall network performance.

In [47], urban events are detected over the data provided from Twitter, and a proactive load balancing mechanism is created by estimating hotspots accordingly. First, data analytics based on Twitter data is applied to design the context-aware module to predict changes in traffic points during events in urban areas. Besides, a proactive load balancing strategy is simulated to automatically configure cell CIOs considering the predicted hotspots. Finally, the strategy is optimized by estimating the best activation time.

The data traffic of vehicle networks, unlike mobile networks, is known to have spatial-temporal regularities due to the periodicity of urban traffic flow. Historical association patterns can thus be used as reference for future traffic flows. Load balancing models created by integrating ML approaches can achieve good correlation solutions by continuously learning from the dynamic vehicle environment, based on historical association experience. The BS is guaranteed to choose the appropriate action, accelerating learning from the similarity between the historical association and the current situation [44].

With the advantage of omnipresent wireless coverage and high-speed data rate, unmanned aerial vehicle (UAV) powered BSs can be deployed in hotspot areas to dynamically meet changing traffic demands and achieve cost-effective deployment. UAV BSs can be deployed intelligently and dynamically based on the estimation of the number of users served in crowded areas where demanded traffic varies greatly. According to the forecast results, the peak hours of user traffic in the predicted areas are determined. Then, UAV BSs can be deployed autonomously and dynamically to hotspots to optimize load balancing [48].

4) FUZZY LOGIC BASED SOLUTIONS

Fuzzy logic (FL) is a set of problem solving methods that provide a simple way to arrive at definitive conclusions from ambiguous and imprecise information. FL deals with reasoning that is approximate rather than fixed and precise. FLC provides a tool that transforms the created linguistic control strategy into an automated control strategy by incorporating the ''experience'' of a human process operator into the controller's design [49]. The main benefit of FLC is the ability to control a system by using linguistic terms such as high or low instead of numerical values when describing the controller. FLC is based on fuzzy set theory, which provides a robust mathematical framework for dealing with ''real world'' and non-statistical uncertainties [50]. The FLC machine consists of four main parts: fuzzification, knowledge base, inference engine and defuzzification processes [51].

- **Fuzzification process:** In the fuzzification stage, the crisp data entries of the system are translated into the FL language. At the end of this process, input values are translated into linguistic terms such as high, medium and low. The fuzzifier acquires the input values and determines the degree to which each of them belongs to the fuzzy sets via membership functions. It thereby transforms the input data into appropriate linguistic values that can be seen as labels of fuzzy sets.
- **Knowledge base:** The knowledge base process consists of two components: a database and a fuzzy control rule base. The database is the control rules created to characterise linguistic variables based on the knowledge and experience of human experts. For this purpose, a set of indefinite IF-THEN rules are defined. It is defined as 'IF' if a set of conditions is met, and 'THEN' when a series of outputs are obtained [52]. The sum of these rules composes the rule base, or rule set, of the FLC.
- **Inference engine:** The inference engine has the ability to simulate human decision making based on fuzzy concepts and to understand fuzzy control actions using fuzzy inference and inference rules in FL.
- **Defuzzification:** A fuzzy set indicating a possible distribution of the control action is transformed into a nonfuzzy (crisp) control action through the defuzzification operator at this stage. The most widely used blurring method is the 'Centroid', which calculates and rotates the centre of gravity of the fuzzy cluster [53].

FLC-based load balancing algorithms are generally designed to adjust a particular network parameter (e.g., CIO) to improve network performance [54]–[56]. A cold start problem does exist since these algorithms initially contain insufficient information and require time to converge. The FLC performance is also limited by the expert knowledge available to it. Unfortunately, increasing the number of rules created by experts is not an effective solution in this scenario. Therefore, FLC algorithms powered by ML approaches have been developed to adapt and improve the FLC rules [57], [58].

5) CHANNEL BORROWING

The load imbalance in the network can be partially alleviated by the fact that congested cells with a high CBR borrow channels from their relatively low-loaded neighbouring cells. Due to the free borrowing of channels, the lender cell eventually becomes the overloaded cell and tries to borrow channels from nearby lightly loaded cells. This process continues with backup. However, channel borrowing-based load balancing strategies cannot be directly applied in 5G/6G wireless networks since cells reuse spectrum bands [59].

D. KEY PERFORMANCE INDICATORS

KPIs are used to evaluate the performance of load balancing models. This section introduces the basic KPIs used or can be used in this regard. In addition, the KPIs used in the papers in the literature are summarized in Table 2.

1) SINR

When associating the user with the BS, unwanted signals are received from other BSs. This interference causes a decrease in the downlink SINR value received by the user. The correct estimation of this metric helps optimise the transmission power for the target quality of service that aids HO decisions, resulting in a more efficient system and a higher perceived service quality by the user. The SINR is formulated as [75]:

$$\Gamma_i =; \frac{P_j G_{i,j}}{\sum_{s \in I/\{i\}} P_i G_{s,j} + P_{noise}} \tag{2}$$

where Γ_i denotes the SINR of user i. P_j and G_{ij} indicate the transmit power of the associated cell and the channel gain from the associated cell to the user, respectively. The first term in the denominator represents the interference power and channel gain of un-associated cells of a user. P_{noise} refers to the power of white noise.

2) PRB UTILIZATION

To accurately represent the load state in cells, it is necessary to correctly measure the cell load. Resource utilisation, measured in PRBs, refers to the usage degree of transmission resources. The PRB utilisation distribution is a useful metric to consider if a cell is experiencing high load during the monitoring period. Let B signify the transmission resources available in the cell (for instance, the number of PRBs), B_u is the average number of resources assigned to user u during the period of interest and U_c is the number of active users connected to cell c. The load of cell c during the period of interest is calculated by the sum of all resources used [76]. The average PRB utilisation of cells is then given as:

$$PRB_{utilization} = \frac{1}{B} \sum_{u=1}^{U_c} B_u; \tag{3}$$

TABLE 2. Paper-based comparison table of KPIs used in performance analysis of ML-based load balancing models.

Paper	Load	PRB utilization	User satisfaction	Throughput	CBR	CDR	OR	PLR
[58]	✓				✓	✓	✓	
[60]			✓			✓		
[61]	✓			✓	✓			
[57]	✓				✓	✓	✓	
[62]		✓						✓
[63]		✓					✓	
[64]	✓				✓	✓		
[65]		✓		✓				
[66]	✓			✓	✓			
[67]				✓				✓
[68]	✓		✓		✓			
[69]		✓			✓			
[70]		✓		✓			✓	
[71]							✓	✓
[72]		✓		✓				
[73]		✓		✓				
[74]		✓		✓				✓

3) USER SATISFACTION

User satisfaction is the measure of how well user requirements can be met. User satisfaction is specifically defined as the probability that the signal quality will be equal to or higher than the specified threshold SINR [77]. User satisfaction is formulated as [77]:

$$p(\Gamma_{thr}) = p(\Gamma_{UEs}) \leq \Gamma_{thr} = 1 - F(\Gamma_{thr}); \qquad (4)$$

Here, Γ_{thr} and $F(\Gamma_{thr})$ denote the SINR threshold value and the cumulative distribution function (CDF), respectively. Through load balancing models, some overloaded cell users are handed over to one or more neighbouring cells, increasing the number of satisfied subscribers. This metric is applied to evaluate the performance of a dynamic network, and the user's QoE is indexed by how many users are satisfied with the service.

4) THROUGHPUT

Throughput refers to the amount of data accurately transferred from one location to another in a given time period. It depends on the current resource availability in the network and how these resources are effectively used [78]. Data throughput rates may fluctuate due to congestion from overloaded cells and inter-cell interference. The HO trigger time decisions of BSs depend on the user's location relative to the cell edge and sudden changes in the radio channel [79]. These throughput fluctuations affect the user's QoE. High throughput is achieved as a result of equal network load distribution among cells. It is defined by the Shannon equation as [80]:

$$Throughput = BW \log_2 (1 + \Gamma_i); \qquad (5)$$

Here, BW stands for PRB bandwidth.

5) CALL BLOCKING RATIO (CBR)

The call acceptance control (CAC) function is responsible for accepting or blocking an incoming call. This function controls the number of free resources available in the candidate cell before making a call-related decision. It prevents users from being handed over to an already congested cell and ensures QoS of both the calling user and existing users in the network. CAC also regulates user access to the network by minimising the number of dropped and blocked calls. CBR is a performance metric that is directly related to call availability and is expressed as the ratio of the number of calls blocked by the admission control to the total number of calls submitted. The CBR is formulated as follows [58]:

$$CBR = \frac{N_{blocked}}{N_{offered}} = \frac{N_{blocked}}{N_{blocked} + N_{accepted}}; \qquad (6)$$

For a call to be accepted, the maximum number of radio resources required must be less than or equal to the number of resources available in the candidate cell. In a load-balanced network, the CBR decreases since there will be unused free resources in the cells.

6) CALL DROPPING RATIO (CDR)

The CDR is the probability that an existing call will be dropped before completion due to poor connection quality. CDR is defined as the ratio of the number of dropped calls to the number of accepted calls. The CDR is formulated as follows [58]:

$$CDR = \frac{N_{dropped}}{N_{accepted}}; \qquad (7)$$

A call may drop in a scenario where available resources are insufficient due to an overload condition other than poor link quality. However, call drops due to the overload condition are assumed to be negligible since sufficient resources are guaranteed for calls accepted by the CAC function. A call is only dropped when the SINR value falls below a certain threshold during a particular time interval.

7) OUTAGE RATIO (OR)

The outage ratio (OR) is another parameter that examines the effect of load balancing on call sustainability. OR occurs from a temporary lack of resources or when the SINR value is below the minimum threshold for a certain period of time. The outage probability of the service provided to a user is defined as [81]:

$$P_{out}^{(t_0)} = \frac{1}{T} \sum_{\tau=1}^{T} P_r(\Gamma^{(t_0+\tau)} - \Gamma_{min}); \tag{8}$$

Since the CRE technique tends to increase interference for cell-edge users, load balancing models that fail to provide efficient interference coordination may increase the outage probability. OR can be formulated as [81]:

$$OR = \frac{N_{out}}{N_{slots}}; \tag{9}$$

Here, N_{slots} refers to the total number of time slots after the HO event is triggered. N_{out} specifies the total number of time slots during which the user SINR falls below the SINR minimum threshold. The CAC acts as a balance between CBR and OR.

8) PACKET LOSS RATIO (PLR)

The packet loss is when forwarded packets fail to reach their destination. PLR expresses the ratio of packets that did not reach their destination to the aggregate number of packets. It is formulated as [82], [83]:

$$PLR = \frac{\sum_{t=1}^{T} \sum_{l=1}^{L} pdiscard_{c\_i}(t)}{\sum_{t=1}^{T} \sum_{l=1}^{L} psizec\_i(t)}; \tag{10}$$

Here, I and T represent the total number of UEs and the total simulation time, respectively. The $pdiscard_{c\_i}(t)$ and $psize_{c_i}(t)$ represent the number of discarded packets and the total number of transmitted packets for cell c, respectively. The PLR increases when a cell does not have enough resources to meet traffic demands. The goal is to increase network efficiency by lowering PLR in cells. This can be achieved by ensuring a fair distribution of resources across the network through load balancing models.

E. LOAD BALANCING PROBLEM FORMULATION

The purpose of load balancing algorithms is to distribute the total network load among BSs in a balanced manner. Several methods have been used to overcome the load imbalance problem, such as using network-wide optimisation techniques [84]. The utility function is widely employed in modelling the user association problem. This function enables the decision maker to quantify satisfaction with a given decision [85]. $U_i(R_{ij})$ refers to the utility function of the ith user, which is the utility of user i when associated with BS j. The utility function $U_i(:)$ is a continuously differentiable, monotonically increasing and strictly concave function [86]. Linear, logarithmic, exponential and sigmoidal utility functions are generally applied in system modelling. The logarithmic function is more commonly used in real systems to provide more resources to users at low rates due to its concave nature (it has diminishing returns) [86]. User association optimisation that maximises utility under resource constraints is mathematically expressed as:

$$\begin{aligned} \max_{x} \quad & \sum_{i \in I} U_i(\sum_{j \in J} x_{ij} R_{ij}) \\ \textit{subject to} \quad & \sum_{j \in J} x_{ij} = 1, \quad \forall i \in I \\ & x_{ij} \in \{0, 1\}, \quad \forall i \in I, j \in J \end{aligned} \tag{11}$$

Here, x_{ij} is the user association matrix. If $x_{ij} = 1$, then user i is associated with BS j (otherwise, it is not). $U_i(R_{ij})$ is the utility of user i when associated with BS j. After examining (11), it has been concluded that the formulation is related to the load of some BSs and the throughput of users associated with these BSs. Users are not always associated with the BS with the highest throughput to maximise the goal of global load balancing. Even if some BSs have high throughput, they can be overloaded and not provide enough resources to users. Therefore, users do not choose these BSs (for enhanced user experience) and associate with low-loaded BSs. This clearly shows that this strategy of maximising the weighted sum of the effective rates can provide a relatively balanced load between BSs [87].

The goal in (11) is to determine the optimum association among all BSs for any given user. However, this optimisation is generally an NP-hard combinatorial problem, since it is assumed that each user can only be associated with a single BS. The necessity to calculate all possible combinations of user associations in solving the global optimisation problem is a very challenging task, even for medium-sized HetNets. A popular way to overcome optimisation complexity is to use fractional user association. The optimisation problem is basically made convex by relaxing the user association matrix from $x_{ij} = \{0, 1\}$ to $x_{ij} = [0, 1]$, and then solving it using convex optimisation tools. However, the optimality of the problem may not be maintained. Using fractional user association in a practical system is also more difficult than the original problem since it requires high coordination and significant message exchanges between users and BSs [84]. Solutions such as classical Lagrangian binary analysis and sub-gradient methods based on convex optimisation are difficult to apply in practice since they are sensitive to algorithm parameters [88]. Therefore, developing load balancing solutions using ML have become more popular in recent years.

F. CCO AND LOAD BALANCING

Among the use cases of self-organising networks, the CCO and load balancing functions are highly important in ultra-dense HetNets. The main objective of the CCO function is to solve issues such as coverage holes, weak coverage, pilot pollution (referring to the interference effect that occurs when at least two neighbouring cells meet the sufficient condition to become the target cell), overshoot coverage and the downlink/uplink channel coverage mismatch by adjusting BS parameters such as the downlink transmit power, antenna tilt and antenna azimuth [89]. Load balancing is a function that has common interests with CCO. The BS antenna tilt is a powerful tuning parameter used in cellular network optimisation thanks to its coverage shaping ability and interference control. The CCO can reduce or expand the coverage of a serving cell through changes in the antenna tilt. This function can be potentially used for load balancing purposes since it can change the service cell of the user.

Figure 4 shows the effect of the change in antenna tilt on CCO and load balancing. Of the two adjacent cells, Cell A is overloaded due to the large number of associated users, while Cell B is relatively underloaded. In this case, Cell A's antenna is tilted down, reducing the coverage area, as shown in Figure 4. However, this process causes a network coverage gap for edge users who were initially in the coverage area of Cell A. They cannot receive service from any cell since they fall in the new coverage gap situation. In this case, Cell B tilts up its antennas to serve users in this area and compensate for the coverage gap. This action sequence allows edge users in an initially congested cell to be handed over to a less dense cell, thus balancing the load in adjacent cells. It should be noted that coverage gaps may still occur in the network if the antenna tilt is not properly adjusted.

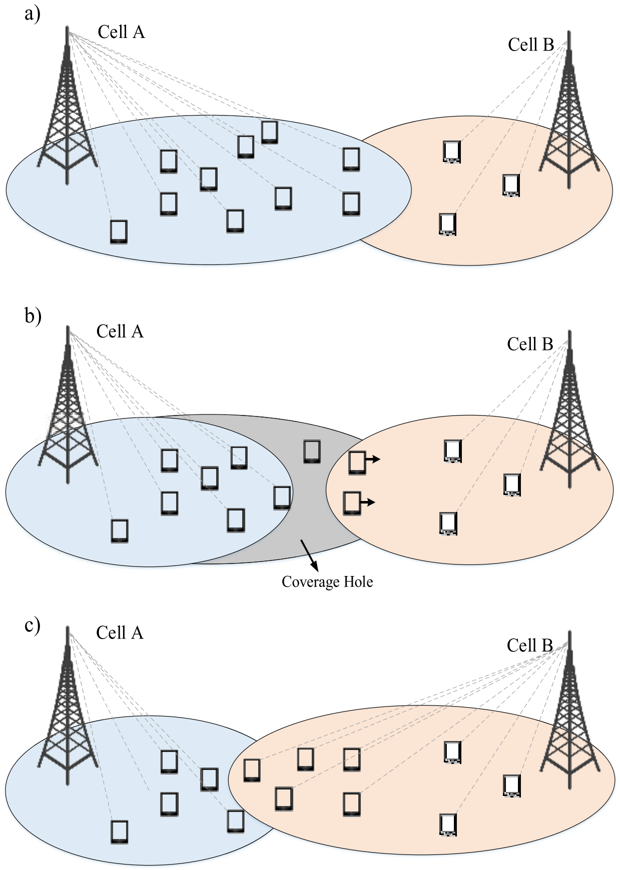

FIGURE 4. Effect of Change of Antenna Tilt, (a) there is a load imbalance between neighboring cells, (b) cell A's antenna is tilted downwards, reducing coverage, (c) cell B's antenna is tilted upwards to widen coverage, thus eliminating coverage holes in the network.

CCO and load balancing functions, if designed and deployed correctly, can enhance user QoE and tremendously increase resource utilisation efficiency in HetNets. A solution where load balancing and CCO functions work together must also consider the CIO, antenna tilt and transmit power parameters along with their interactions. In [90], load balancing and CCO techniques are jointly applied. The solution uses three key parameters (BS transmit power, antenna tilt and CIO values) in a single formulation for the optimisation process. The data obtained from the simulation indicates that the proposed solution does enhance the throughput, spectral efficiency and load distribution. In [91], antenna tilt and CIO values are adjusted to jointly optimise CCO and load balancing.

G. COORDINATION BETWEEN MRO AND MOBILITY LOAD BALANCING

MRO and mobility load balancing are two important functions that automatically optimise network performances. The MRO minimises HO problems, and mobility load balancing basically balances the loads between cells. Both functions optimise network performance by adjusting HO parameters such as CIO, *Hys*, time-to-trigger (TTT), etc. Although the two functions work independently, there is a close relationship between them. Conflict can occur when the MRO function and the mobility load balancing function change the same or related HO parameters in the opposite direction or towards the same direction but at different scales. This conflict significantly wastes network resources and does not improve performance. This problem cannot be avoided unless either the MRO function or the load balancing optimisation function is suspended for a period of time.

1) OPERATIONAL PRINCIPLE OF MRO

The MRO collects information on UEs at a given interval after detecting HO problems. The purpose of the MRO is to adjust a cell's *Hys* and TTT to select the optimum HO trigger point and simultaneously check for redundant HOs and radio link failure (RLF). RLF severely impacts the user experience since it causes the UEs to HO from the serving BS, ensuring the continuity of the mobile connection. The main cause of RLF is the RSS of the UE's serving cell is too low and interference is too high [92]. Improper HO triggering causes RLF to occur. If HO is triggered when the RSS from the target cell is too low (defined as too early HO), the RLF will occur shortly after the start of the HO procedure. This is because the user's connection quality is low, therefore, the UE is associated with the source cell again. When HO is triggered much later than the appropriate time (defined as too late HO) due to the RSS from the source cell being too low, the RLF occurs in the serving cell before or during the HO procedure, and the UE is associated with a different target cell than the serving cell. Due to RLF, the UE physically loses its radio connection to the BS, causing additional retransmissions or reconnections.

This results in service interruptions provided to the UE and wastage of network resources [93]. The PPHO is the back and forth HO of the UE between two neighbouring cells within a short period of time, also called the minimum state time. In PPHO, no decrease in QoS is provided to the user, but additional signalling between the UE and the BSs during the HO procedure results in wasted network resources.

2) OPERATIONAL PRINCIPLE OF MOBILITY LOAD BALANCING

Mobility load balancing regulates the cells' coverage areas by adjusting the HO parameters to deal with uneven traffic loads between cells. It periodically monitors cell loads and accordingly adjusts their CIO when faced with a load imbalance. Transferring cell edge UEs from highly loaded cells to lightly loaded neighbouring cells will increase resource utilisation efficiency, thereby reducing the call blocking ratio. For instance, if cell 1 is heavily loaded and neighbouring cell 2 is lightly loaded, cell 1 can increase the $CIO_{1\rightarrow 2}$ so that HO is triggered earlier than usual. Thus, UEs moving from cell 1 to cell 2 are handed over sooner since they meet the necessary condition for HO and the load on cell 1 is reduced.

3) THE CONFLICT

Although MRO and mobility load balancing work independently, they are interrelated since both set the HO parameters for the optimisation process. Both functions optimise network performance by adjusting HO parameters such as CIO, *Hys*, TTT, etc. Conflicts can occur when these two functions change the same or related HO parameters in opposite directions or towards the same direction but at different scales. Consider the scenario where cell 1 is heavily loaded and cell 2 is lightly loaded. The $CIO_{1\rightarrow 2}$ will increase to make it easier to HO from cell 1 to cell 2 for load balancing purposes. However, this operation of handing over UEs from cell 1 earlier reduces the value of the input condition for cell 1's A3 event. This will lead to RLF since it will unfortunately cause a too early HO. Mobility robustness optimisation reduces the $CIO_{1\rightarrow 2}$ to minimise the number of too early HOs from cell 1 to cell 2. However, the load of cell 1 is still very heavy, so mobility load balancing will increase $CIO_{1\rightarrow 2}$ and MRO will again reduce $CIO_{1\rightarrow 2}$ due to HO issues. This conflict becomes a vicious cycle. Two consequences occur from this situation. First, the network becomes highly loaded for a long time, decreasing the efficiency of mobility load balancing. Second, it can reduce UE satisfaction and waste network resources due to higher CBR and CDR.

H. CONTROL OF LOAD BALANCING ALGORITHMS

Load balancing strategies are divided into three groups: distributed, centralised and hybrid. This section describes each strategy.

1) CENTRALIZED LOAD BALANCING TECHNIQUES

In the centralised approach, the network has a centralised control node that performs resource allocation. The centralised node collects load performance metrics (e.g., available PRB) or notifications (e.g., threshold crossing of certain metrics) and analyses the load metrics. Based on the information obtained, it can update the HO and/or reselection parameters of the cell or its neighbours to optimise traffic load distributions between cells. The centralised load balancing technique can provide optimum resource allocation for the entire network since it has a global perspective and can make quick decisions. However, the required signalling overhead for medium to large networks can be excessive [85].

2) DISTRIBUTED LOAD BALANCING TECHNIQUES

In this technique, no centralised control node is present to monitor the network and make load balancing decisions as a result of evaluations. Instead, this technique enables each node in the network to autonomously transfer its load to neighbouring underutilised nodes to achieve load balancing. The load balancing decisions made by the nodes are based on their own observations from the network. It is particularly suitable for use in large HetNets. However, when nodes make decisions to optimise their returns in their own interests, this can result to an inefficient global use of network resources.

3) HYBRID LOAD BALANCING TECHNIQUES

These techniques combine the advantages of both centralised and distributed load balancing techniques.

III. ML IN LOAD BALANCING

This section provides a comprehensive analysis of ML-based load balancing models. In the Section III-A, a preliminary explanation is given about the ML algorithms used in the literature. In the Section III-B, ML-based load balancing models in the literature are shown item by item. Each item represents an article (with publication dates included). This section provides the reader with a historical flow of articles published in this field since 2013. Thus, the development of ML-based load balancing models over time can be observed more easily. Here, a detailed examination has been made about the structure of each model, the steps followed in its design, the analysis of the models and their deficiencies. The aim is to highlight the points to be considered while creating new designs. Table 3 contains a summary of the studies reviewed in this section. This will simplify the comparison process when examining models.

A. AN OVERVIEW OF ML ALGORITHMS

ML is a collection of methods that allows computers to learn, automate, and optimize a model that helps find patterns [94]. An ML approach consists of two phases: the training phase (the system model is learned through the training data) and the decision phase (an estimated output is generated for each new input through the trained model). Depending on how learning is made, ML approaches fall into three basic categories: supervised, unsupervised and reinforcement learning (RL) [95]. In the following, different ML algorithms are briefly described under these three basic categories.

TABLE 3. Comparison table for all ML-oriented load balancing studies throughout the literature.

Paper	ML techniques	Learning model	Efficiency	Network type
Munoz et al. [58]	FLC combined with RL	• Examines the potential of using solutions based on only FLC, as well as FLC coupled with Q-Learning, to self-adjust femto cell transmission powers or HOMs and solve localised congestion problems.	• Q-Learning performs better when applied to transmit power rather than HOM.	HetNets
Kudo et al. [60]	RL	• Each UE learns the optimal CIO value from historical experience through Q-Learning.	• Reduces the OR and improves the network throughput compared to the scheme using the optimum common bias value.	HetNets
Xu et al. [61]	RL	• Load status of cells are predicted through Q-Learning. • HOMs and transmission powers of target cells are adjusted according to the predicted load state.	• Increases the throughput of the system, improves the CBR and relieves congestion.	HetNets
Munoz et al. [57]	FLC combined with RL	• A unified self-management mechanism based on FL and RL was proposed for the relationship between load balancing and MRO. • The FS adjusts the cell's HO parameters and the RL optimises it so that the most appropriate action for load balancing or MRO can be selected.	• Partially alleviates congestion and reduces the number of HOs compared to reference states where load balancing and MRO operate separately or even simultaneously in an uncoordinated manner.	Macro-cellular environment
Franco et al. [62]	Multivariate polynomial regression	• Uses a polynomial regression method to learn its parameters, then uses its model to adjust the CIO values.	• Balances the traffic load and reduces packet losses compared to HetNet models without CRE and homogeneous networks.	HetNets
Simsek et al. [115]	RL	• Learns to optimise traffic loads and CIO values of BSs using RL. • User programming is performed based on the speed of each UE as well as historical data rates exchanged between tiers.	• Increases the UE throughput and significantly reduces the possibility of HOF.	HetNets
Semov et al. [116]	Autonomous ML model	• The load balancing problem is made into small tasks that can be solved with ML techniques.	• System capacity can be significantly improved.	HetNets
Ali et al. [63]	Game theory	• CRE-based approach for load balancing and ABS for interference management are used. • User association is formulated as a potential game and the proposed game must be solved using log-linear and binary-linear learning algorithms.	• Outages are reduced and better load balancing is achieved.	HetNets
Li et al. [44]	RL	• In the RL-based initial learning model, the initial vehicle-BS association decision is made based on available context information only. • After a certain learning period, when the BS encounters network changes again, it identifies new associations using the RL method based on historical association patterns.	• Compared to max-SINR and 3D schemes, the proposed model achieves the minimum load variance of multiple cells. • Has higher proportion of large overall service rates.	Vehicular networks
Musleh et al. [64]	RL	• Load balancing algorithms based on the RL of end-user SINR and the macro cell throughput attempt to find the policy that determines the optimum power level for the femto cell using the Q-Learning method.	• Improves load distribution and provides lower CBR and CDR rates for highly loaded macro cell.	HetNets
Shahriari et al. [117]	DRL	• A DRL-based GOL system was proposed for load balancing in C-RAN. • Load balancing is accomplished using GOL, determining the best possible VMs in the BBU pool to assign users.	• Reduces both cache-misses and communication load.	C-RAN
Huang et al. [88]	CE based algorithm	• Using a CE algorithm, the user association is optimised to approximately maximise the network utility. • An almost optimal association is achieved by iteratively refining the probability distributions through CE minimisation.	• Provides close to optimal performance in terms of utility rate and load balancing.	HetNets
Gomez et al. [118]	PCA, Extra trees, RL	• A load balancing scheme based on ML techniques using both unsupervised and supervised methods and the MDP was proposed.	• Improves network capabilities in terms of packet delivery rate and energy cost of data delivery.	IoT Networks
Hu et al. [48]	Hybrid ARIMA-XGBoost	• In the smart and dynamic distribution plan, according to the predicted number of service users in hotspots of the UAVs, linear predictions are made with the ARIMA model. Next, the non-linear XGBoost model is applied to the ARIMA residue.	• Improves the capacity and QoS of the network	Small cell networks
Roshdy et al. [65]	SVM	• Using the SVM method, the cells' current resource usage is estimated and their CIO values are adjusted to provide load balancing.	• Increases resource utilisation and user throughput.	Homogeneous network

TABLE 3. ***(Continued.)*** **Comparison table for all ML-oriented load balancing studies throughout the literature.**

Paper	ML techniques	Learning model	Efficiency	Network type
Li et al. [119]	DL, NLP	• A load balancing technique was proposed for efficient coordination between mobility edge servers. • A deep CNN is used to learn from spatio-temporal features to predict the traffic situation. • The load balancing problem is formulated as an NLP problem and a new framework for NLP optimisation is adopted in CNN.	• Copes with the unstable QoS experienced by different MBSs and im proves network resource utilisation.	Vehicular ad hoc network
Barros et al. [120]	DRL	• The Gaussian process is used to predict the load of a node, and DRL is applied to balance the load of the network.	• The proposed model supports a convenient trade-off between load balancing performance and speed.	D2D networks
Baek et al. [121]	RL	• The SDN fog controller chooses the optimum offloading decisions using the greedy algorithm and Q-Learning. • According to the status information of fog nodes, the number of tasks is determined to transfer to the neighbouring node.	• Provides lower average processing delay and lower probability of failed allocation.	Fog networks
Xu et al. [66]	DRL, k-Means	• DRL-based load balancing is proposed along with a two-tier architecture. • The top tier selects the most overloaded small BSs as the initial cluster centroids, then dynamically groups the small BS into clusters using a customised k-Means algorithm that groups the small BSs by location. • The lower layer balances the load distribution within the cluster using a DRL-based algorithm. • With the offline safeguard mechanism, optimum operation of the online system is ensured.	• Performs better than reference algorithms in terms of load balancing. • Achieves good scalability over the number of small BSs. • Safeguard mechanism ensures that online performance is stable and not degraded.	UDN
Huong et al. [122]	DQN	• In a NN-based two-component structure, the policy estimator predicts the policy and based on the forecast results, the policy maker updates its knowledge to improve policies.	• Improves latency and network utilisation.	SDN networks
Yao et al. [67]	NN, DL	• An ML-powered load balancing routing scheme was proposed using network state information to train the NN and create route predictions.	• Reduces PLR and queuing delay.	Packet-switched networks
Mohajer et al. [68]	DRL	• The learning automata sends each request to the servers at a rate. This rate is updated by training the model over the average response time of servers.	• Reduces the HOF rate and the service blocking rate compared to the adaptive load balancing with ICIC and constant mobility load balancing models. • Reduces the number of dissatisfied users.	Cellular networks
Attiah et al. [69]	DRL	• Decisions are made over CIO values as a steering action using DRL to provide network load balancing.	• Improves network DL efficiency and absolute deviation of PRB utilisation compared to the model with zero CRE offset.	Cellular networks
Ahmad et al. [70]	RL	• RL is trained to determine the most optimal policy using the trust region policy optimisation algorithm. • The optimal policy determined is used to estimate the appropriate access point assignment for the particular situation.	• Exhibits similar performance to the exhaustive search scheme in low complexity. • Outperforms the signal strength strategy scheme and iterative algorithm in most scenarios	Hybrid Wifi/Lifi networks
Jialing et al. [71]	RL	• Two Q-Learning-based selection strategies were combined to compensate for the load and energy imbalance. • In the downlink, the optimal CIO values that provide load balancing are chosen. In the uplink, the best routing target according to the energy status of neighbouring BSs is chosen.	• The number of outage UEs is reduced and longer network life is provided.	HetNets
Lei et al. [124]	DRL	• A DRL-based multitasking control model was proposed, with the congestion control as the main task and load balancing as the auxiliary task.	• Shorter and better RTT values are obtained.	SDN-based networks
Fan et al. [72]	DL	• With DL, future traffic routing strategies are predicted from past traffic offloading results.	• Improves network throughput, load balancing performance and user service rate.	Cellular V2X Networks
Alsuhli et al. [73]	DRL	• Provides a DRL-based load balancing model that adjusts the CIO values and transmission powers of cells.	• Compared to the model that only adjusts CIO, the load balancing performance is similar but better in terms of throughput. .	Homogeneous network
Madelkhanova et al. [126]	RL	• Q-Learning is used to learn the appropriate adjustment of CIO values according to the load of macro BSs.	• Increases the capacity and satisfaction of UEs.	HetNets

TABLE 3. ***(Continued.)*** **Comparison table for all ML-oriented load balancing studies throughout the literature.**

Paper	ML techniques	Learning model	Efficiency	Network type
Mai et al. [125]	RL	• A hybrid intelligent in-network load balancing scheme using multi-agent actor-critic RL was proposed. • Centralised learning and distributed execution framework has been adopted. • The centralised "critic" is reinforced by the joint actions of all representatives, and distributed switches act according to their local observations.	• Increases convergence speed and performance.	Data center networks
Asghari et al. [127]	RL	• Q-Learning and SARSA based load balancing algorithms that adjust the CIO value of cells were proposed.	• Provides better load balancing performance compared to models using fixed CIO	Homogeneous network
Iturria et al. [74]	DRL	• Determines the CIO value of each cell through a central agent using the CDQL algorithm.	• Improves throughput, latency, jitter and PLR performances.	Homogeneous network

1) SUPERVISED LEARNING

In supervised learning, the agent is given samples of labelled state-action pairs, along with an indication that the action is ’right’ or ’wrong’. The basis of supervised learning is to build a general policy from training samples. In this way, the system’s responses are predicted or generalised to behave correctly in situations not included in the training set. The most common algorithms used in supervised learning are presented below:

a: REGRESSION

Regression is a statistical method used to model the relationship between independent variables (inputs) and dependent variable (outputs) in the form of parametric equations. It helps us to understand how the value of the dependent variable corresponding to an independent variable changes when other independent variables are kept constant, based on the standard error estimates provided by the modeling paradigm. Variables in the regression model are continuous. In this method, the independent values are found first. Then, the coefficients of the independent variables are calculated so as to minimize the differences between the actual and estimated values. Finally, the formula is made ready by adding possible random errors. The regression types frequently used in the literature are linear regression and polynomial regression. Detailed information about the regression algorithms can be found at [96].

b: SUPPORT VECTOR MACHINES (SVM)

The SVM simply takes a set of inputs to be classified as points in a high-dimensional space and tries to find a line separating those points. The distance from the hyperplane separating the two classes to the nearest expression vector is defined as the margin of the hyperplane. SVM tries to maximize the margin separating the hyperplane in order to maximize its ability to predict the classes of unclassified instances in the feature space. In the absence of a good linear separator, data is projected into a higher dimensional area with kernel function techniques. However, with increasing dimensionality, the number of possible solutions increases, and classes may become instance-specific in the training dataset, which cannot provide a general solution to a new input. We recommend that readers who want to get more detailed information about SVM algorithm take a look at [97], [98].

c: DECISION TREES (DT)

DT classifies data items by asking a series of questions associated with each internal node’s attribute of the items. Each internal node is divided into sub-nodes for each possible response to be given, thus creating hierarchy coded as a tree. An unlabeled instance is classified according to the valid responses by following the path from the top node to a root childless node. In DT, the information in the lower branches is more pure than the information in the upper branches. One point that should be taken into consideration in the application of DT is to limit the complexity of the learned trees so as not to exceed the training examples. There are two techniques that are generally applied: to stop the split when no question increases the purity of subsets by more than a small amount, and to prune the tree by deleting the nodes to prevent the training data from overfitting. We recommend that readers who want to learn more about the DT algorithm take a look at [99], [100].

d: NEURAL NETWORKS (NNS)

NNs can be defined as a highly complex, non-linear, parallel interconnected network of basic computational elements that display information processing properties similar to several hypothetical models of the brain’s functioning. In NNs, the equivalent components of neurons in the human brain are nodes. Nodes are interconnected with variable link weights and are responsible for nonlinear calculations. Generally, sigmoid or hyperbolic tangent functions are used as activation functions [101]. NNs consist of three layers: input layer, hidden layer and output layer. NNs have an input and an output layer, but the number of hidden layers is not fixed. For complex models, the performance of NNs can be improved by optimizing the number of hidden layers and nodes in each layer (improving the ability to learn nonlinear relationships between input and output). An error function can be defined as the difference between the node output and the target output, and the weight vector is updated at each step with the help of the adjustment ratio to converge the system. We recommend that readers who want to learn more about the NNs algorithm take a look at [101], [102].

e: K-NEAREST NEIGHBOR (K-NN)

K-NN is a supervised learning technique in which the class of an unclassified data sample is determined by the closeness of a set of previously classified points. This algorithm can be applied to problems where the common distribution underlying the observation and result is unknown or difficult to determine [103]. In the k-NN algorithm, if most of the k-NNs of a new unclassified instance belong to a particular class, the sample will be classified into that class. For higher values of K, the effect of noise decreases in classification and the robustness of the model increases. The performance of the K-NN classification depends on how distances between the unlabeled sample and its nearest neighbors are calculated. When prior information is not available, most k-NN applications use simple Euclidean distances to measure the difference or similarity between two states. In [104], [105], some suggestions related to this issue have been made in order to increase the performance of the k-NN algorithm. Readers who want to get more detailed information about k-NN can refer to [98], [104], [106].

2) UNSUPERVISED LEARNING

Unsupervised learning algorithms aim to discover patterns, structures or information from unlabelled input sequences without having a controller that provides the correct answers or grade of error for each observation. In the following, most common algorithms implemented in the supervised learning are presented:

a: K-MEANS

K-Means is an algorithm used to classify or group a set of unlabelled data by attributes and features into K numbers of groups. Only the initial dataset and the desired number of clusters (K) parameters are sufficient to implement this algorithm. The algorithm is simple and the following steps are followed: (a) random K number of centroids are determined to cover all points in the dataset; (b) all remaining points are assigned to the nearest centroids using a distance function; (c) because the centroid location is not exactly correct, the new centroid is determined according to the updated data and all data is assigned to this new centroid; (d) it is repeated until the condition for convergence is met, that is, until no data is moved to another cluster anymore, otherwise it returns to (b). We recommend that readers who want to know more about the k-Means algorithm should take a look at [98], [107].

b: SELF ORGANIZING MAPS (SOM)

SOM is essentially an unsupervised NNs model. The SOM algorithm is used in dimension reduction and data clustering applications. A SOM consists of a grid of neurons, and inputs are automatically associated with the nodes of a two-dimensional grid on a regular basis, such that more similar models are automatically associated with adjacent nodes in the grid, while less similar models are placed farther apart in the grid [108]. Thus, this algorithm, which is a kind of similarity diagram, provides an idea of the topographic relationships of complex, nonlinear original data. Each neuron in the model has a weight vector, and after a sample is fed into the system, a distance function is used to calculate the similarity between the input data sample and all weight vectors to determine which neuron is closest to the sample. The neuron to which the input is nearest is called the best matching unit, and an unknown input item is then classified according to that node. We recommend that readers who want to get more detailed information about the SOM algorithm should look at [108], [109].

3) REINFORCEMENT LEARNING

RL includes learning to match states to actions by attempting to maximise a numerical reward signal. In RL, the agent is not told what actions to take (unlike other ML techniques). Instead, the agent is expected to discover by experimenting with actions that will provide the most rewards. The three most important distinguishing features of RL problems are that they do not have direct instructions on which actions to take, being closed-loop problems (the actions of the learning system affect the inputs in the next steps), and the consequences of actions continue to have an effect over long periods of time (actions affect not only the current reward but also the next state and therefore all subsequent rewards) [110]. The RL system consists of four main elements: a policy, a reward signal, a value function and an environment model.

- **Policy:** The policy defines how the learning agent should behave at a given time. Basically, a policy is responsible for matching states observed from the environment with actions taken by the agent. Policies are generally stochastic and optimal policy is defined as the policy that produces the largest cumulative reward over all states [111].
- **Reward signal:** A reward signal identifies the goal in the RL problem, and the agent's goal is to maximize this cumulative reward in the long run. The environment sends a reward to the learning agent at each step providing an assessment of the current state. The reward signal identifies what good and bad events are for the agent based on the consequences of the previous action. The agent cannot directly influence the reward signal, but can do so indirectly by changing the state of the environment. It uses a reward-punishment system. If a low reward comes after an action chosen by the policy, the policy can be changed in the future to choose another action in this state.
- **Value function:** The reward signal instantly shows what is good, while a value function indicates what is good in the long run. Basically, the value of a state is the

total amount of rewards a representative can expect to accumulate in the future, starting with that state.

- **Environment model:** An environment model is something that mimics the behavior of the environment or allows inferences about how the environment will behave. The reason for creating the environment model is planning. Basically, given a state and an action, the environmental model enables a plan of action to be decided by considering possible future states before actually experiencing them.

One of the issues to consider in RL is the trade off between exploration and exploitation. In order to increase the amount of reward achieved, the representative should choose actions that have been tried in the past and produced a lot of rewards in the long run. The agent should explore the environment by performing previously untested actions and perceiving their consequences. Because the agent has to leverage the knowledge it has to receive rewards in order to provide far more rewards than any other action being tested, it also has to research to make better action choices in the future. But the exploration process does not guarantee better performance because actions may be less good than current policy.

We recommend that readers who want to have more insightful information about RL algorithm look at [110]–[113].

B. RELATED STUDIES BASED ON ML

Although supervised learning is an effective technique, it can be difficult to acquire training data from the field for load balancing problems. In the literature, one of the solutions for obtaining measurements, which should be provided from an operational network in the application of supervised learning, is to create training data from simulation programs. However, the reliability of predictions in such a solution would depend on the correctness of simulations and the quality of measurements. Another solution is to apply previous real-world datasets as a training set. The main challenge of this solution is that the planning strategy created with past traffic flow observations must be performed at the very beginning of the next time interval. In supervised and unsupervised learning, data is generally considered as static, therefore, performance is measured according to the dataset given to the system. In the RL perspective, data is considered as a moving target, meaning the learning process is driven by current policy yet this policy may change according to the distribution over states and rewards. This ML technique is popular since RL can be successfully applied to states where there is no traceable action model. It can also learn something from its own experience through a representative in the undiscovered region where learning is expected to be most beneficial. Deep reinforcement learning (DRL), where RL is blended with deep learning (DL), is an important solution in handling the large expansion of the state-action space created from the RL application in complex network scenarios. This contribution highlights DRL from recent studies throughout the literature.

The SDN architecture centralises network intelligence at the application layer and control layer instead of distributing network intelligence among network devices, as in traditional network architectures. Since the SDN controller has a global view of the network, it has become the leading architecture in the creation of load-balanced routing schemes in recent years. The centralised SDN controller constantly monitors the state of the network, making it ideal in load balancing models, especially in fog computing where ML algorithms require a significant amount of data for accurate decision making [114].

In the following, examination of each ML-based load balancing algorithm in the literature since 2013 are presented:

- **In 2013 [60]**, a Q-Learning based scheme was proposed for the optimisation of each user's CIO value. Each UE learns the optimal CIO value from historical experience, optimising the number of outage through Q-Learning. The proposed algorithm has a storage issue since all users must store the Q-values of the CIO values in the Q-table. Dimensionality is therefore a significant hurdle for this model, with little scalability to add other cell types. The large memory size required for learning does not allow convergence in a practical time. Although the algorithm takes into account RB allocation as a measure to balance the traffic load and minimise interference, no clear measurement has been made to ensure the balance between metrics. Furthermore, the effect of user mobility is not considered in the algorithm. The data obtained from the simulation indicates that the employed method can reduce OR and enhance throughput compared to models using a common CIO.
- **In 2013 [58]**, the potential of various load balancing techniques was explored based on the self-adjustment of femto cell transmission powers or HO margins (HOMs) to solve localised congestion problems in femto cells. This work used solutions based on FLC alone as well as FLC combined with Q-Learning. The key contribution of this work is the proposal for a load balancing mechanism that combines both the fast response of FLC and the performance improvement of RL systems. Another important feature is that it does not require any available priori information to adjust the behaviour of the system, thereby adapting to any changes that may occur within. However, the solutions in this study disregarded the limitation of femto cell processing capacity. It was designed to solve local and persistent congestion issues while ignoring the difficulty associated with temporary congestion issues.
- **In 2014 [61]**, a Q-Learning algorithm was proposed to solve the cell congestion problem by estimating the load state of each cell. This algorithm is based on adjusting HOMs and transmission power of target cells according to the predicted load state of the cells. The advantage of this model is that it can be adjusted to new behaviour in the data. However, the presented model lacks online learning and does not tolerant input noise.

- **In 2015 [57]**, a self-management model based on FL and RL was proposed to achieve joint optimisation of load balancing and MRO functions. The proposed model is based on a fuzzy system (FS) that adjusts HOMs at the neighbourhood level of cells to improve network performance by jointly enhancing these functions. FS is optimised by the Q-Learning algorithm which guides the selection of the most appropriate action to meet load balancing objectives and MRO functions. The decision regarding which action FS should take is made by learning from its past actions and its influence on network performance. This paper dealt with both fast and slow users. The data obtained from the simulation indicates that the model performed better than the standalone entities that simultaneously operate in the network.
- **In 2015 [62]**, a self-optimising CRE scheme based on a statistical learning approach for HetNets was proposed A polynomial regression method is applied to learn the parameters, and the model then adjusts the CIO values. Small cell cover age area is dynamically expanded according to traffic conditions. However, the solution is insufficient in evaluating the impact of CIO on user QoE. The relationship between parameters can be complex and may depend on several parameters. In complex scenarios, the polynomial regression method may not perform well. These algorithms lack information about user behaviour, such as mobility and preferences, since the learning process is based only on cellular data. Such solutions are problem-specific and do not consider conditions where traffic demand is unusually high, such as popular events.
- **In 2015 [115]**, a context-aware mobility management procedure for small cell networks was proposed. The scheme improves HO performance and throughput using RL techniques and intercellular coordination. The procedure proposes short-term and long-term solutions. In the long term, optimisation of CIO values of small cells is learned using RL techniques to achieve load balance. In the short term, user scheduling is conducted according to the speed of each user and the data rates exchanged between tiers. The collected data from the simulation indicates that the approach improves throughput and HOF probability performances over the traditional mobility management method.
- **In 2016 [116]**, a distributed autonomous load balancing solution using a programmable learning model was proposed. The model essentially abstracts the complex task of load balancing through existing pre-processed data. It attempts to solve the complex task using ML techniques, transforming it into small tasks with modularity and adaptation approaches. The CIO values of cells are dynamically adjusted according to the HOF and CDR performances obtained from the HO recordings. However, the model's passion for computation and the need for strong coordination between autonomous processes prolong the time required for the learning phase.
- **In 2016 [63]**, a CRE-based approach for load balancing and ABS for interference management were employed. The approach formulates the user relationship as a potential game. The linear learning algorithms are used to solve the game and reach pure Nash equilibrium. In such optimisation algorithms, key parameters of the model must be collected from the network. However, due to simplifications and/or assumptions, the optimal configuration of the model differs from the optimal configuration of the network [37].
- **In 2017 [44]**, an online RL scheme was proposed for balancing loads in vehicle networks. Two RL stages are present. In the first learning model, the association problem between vehicle and BS is formulated as a multi-armed bandit problem and the initial association decision is made by RL based on the available context information. The second learning phase is called historical RL. In this stage, the spatio-temporal regularity of vehicle networks is utilised. The aim of the model is to balance the load in dynamic environments based on historical patterns in the initial learning stage. Each BS is an agent. An association matrix is created by calculating the similarity between the current environment and the historical models. The proposed model provides higher service rates and improved convergence time compared to traditional max-SINR and distributed dual decomposition optimisation schemes. However, interference was not considered, and the spectrum allocation and transmit power control of each vehicle and BS remained unexplained.
- **In 2017 [64]**, two RL based algorithms were proposed to balance traffic loads. Both algorithms use the Q-Learning technique to learn the optimal policy for the best power levels of femto cells. RL load balancing of end-user SINR monitors the SINR of UE in macro and femto cells. It also monitors their CDR and CBR and adjusts the transmit power of femto BSs. Load balancing based on the RL of macro cell throughput mainly considers cell throughput for all UEs instead of average SINR. It observes the results of actions on the average cell throughput, CBR and CDR as well as updates the Q-table to obtain constant throughput. The two algorithms were compared with the fixed reference signal power allocation method. The data obtained from the simulation indicates that the charge distribution does improve, providing lower CBR and CDR for the highly loaded macro cell.
- **In 2017 [117]**, a DRL-based general online learning (GOL) system was proposed for load balancing in cloud radio access network (C-RAN). GOL has a hierarchical structure consisting of three parts: a medium and high entity as well as numerical and generic entities. As input in the first layer, the last input data and previously executed output data are entered into the system as a stream in real time. The numerical entity is stored in the first layer, and the historical data for

the medium entity is stored in the second layer. The high entity is stored in the third layer. There are two channels in GOL: the input channel which detects its environment based on the input data, and the output channel which allows it to interact with the environment. Load balancing is achieved by associating users with the best possible virtual machines (VMs) in the baseband unit (BBU) pool through the GOL algorithm. The main task of the load balancer is to minimise the total cache losses and signalling load between the VM cache and cloud storage. The simulation data informs when the GOL scheme provides significantly good performance in reducing cache losses and signalling overhead.

- **In 2018 [88]**, user association was optimised using a cross-entropy algorithm to approximately maximise network utility. The original problem was initially formulated as a cross-entropy (CE) minimisation issue with the aim to learn the probability distribution of variables in the optimal relationship. The proposed approach solves the combinatorial optimisation problem more easily compared to the typical relaxation techniques, thanks to the adaptive update procedure. To solve this formula, a stochastic sampling method was presented. The algorithm first generates random samples according to the assumed probability distribution. It then selects the best samples as ''elites'' by calculating the total utility ratio of each sample in the problem. The probability distribution parameter is subsequently updated according to the elites selected by minimising CE. At each iteration, the CE approach increasingly concentrates around the optimum design by generating a sequence of sampling distributions. The data obtained from the simulation indicates that this algorithm provides better load balancing than the max-SINR algorithm. Compared to available solutions based on convex optimisation, this approach is not sensitive to algorithm parameter choices, which means that the proposed approach may be more efficient in practice.
- **In 2018 [118]**, a load balancing model based on the ML technique using the Markov decision process (MDP) as well as unsupervised and supervised learning was proposed for an urban IoT network. Data is preprocessed because historical data from a real operating network includes raw data. Accordingly, useful entries are cleaned and selected. Thus, for each BS, some features are extracted from the data and some measurements are waived. After pre-processing the existing data, BS samples are analysed using the principal component analysis (PCA) method to ascertain whether the extracted features provide differentiated models for each BS. A supervised classifier is used to estimate the BS that should transmit downlink messages to an end device using variables not directly related to signal strength. The proposed model learns from the data to predict a relationship between device and BS without considering signal-based measurements. MDP is also used to determine whether the BS loads should be balanced. It improves the packet delivery rate by reducing communication costs, such as the amount of energy required for packet delivery. However, the implementation of such complex models requires extra care with sufficient resources. The model's limitation is the time delay that the decision process can cause. The time complexity analysis should also be considered, especially when many end devices are present.
- **In 2018 [119]**, an end-to-end load balancer was proposed to provide efficient load balancing for I2V communication by adjusting mobile end servers according to road traffic situations. The proposed load balancing model consists of two main parts. In the first part, convolutional neural network CNN) is applied to predict the state of road traffic based on historical road traffic information and to learn the spatio-temporal correlation. In the second part, the load balancing problem is formulated as a nonlinear programming (NLP) problem. A CNN-based framework is also used to approach NLP optimisation and schedule the caching and transmission of high-resolution maps based on predicted road traffic conditions. To directly train the two deep CNNs that compose the model, a large number of training datasets and computational resources are required. Therefore, these two deep CNNs are separately trained and then connected to each other.
- **In 2019 [65]**, the CIO parameter was adjusted to increase user throughput by applying the SVM algorithm. The system is trained with different radio attributes, and the SVM estimates user throughput at power and code utilisation value. The cells are then ranked by current and required user efficiency, power utilisation and code utilisation. CIO values are determined for each cell according to the cell rank, the value of the traffic to be offloaded and the cell traffic distribution. However, such solutions are often difficult to adapt to dynamic network scenarios where user traffic consistently fluctuates. HetNet scenarios with multiple frequency bands were not examined, and problems arising from frequent inter-frequency measurements were not considered.
- **In 2019 [120]**, a load balancing model using the DRL algorithm was proposed for device to device (D2D) net- works. This model applies the Gaussian process to estimate the load of a node and applies a DRL to balance the network load. The model was compared to the Robin Hood approach, which does not consider any factors other than the current load of nodes. The data from the simulation indicates that the proposed model does improve the load balancing performance but does not make a big difference in terms of overall performance.
- **In 2019 [121]**, a load balancing algorithm using Q- Learning was proposed in SDN-based fog networks. The proposed algorithm learns the policy of forwarding the desired number of tasks from fog nodes to the most

suitable neighbour node for load balancing purposes. The architecture consists of an SDN fog controller and serving SDN fog nodes. The RL in the SDN controller collects information by globally monitoring the state of the network and determines how many tasks should be forwarded to the target neighbour node. This is based on the size of the request tasks in the fog nodes and the number of tasks remaining in their queues.

- **In 2019 [48]**, an ML-based UAV BS smart deployment scheme was proposed to evaluate performance in a real-world dataset. Data preprocessing and data analysis is done on raw data to improve data quality. With the conditional mean imputation method, the missing values in the dataset are filled according to the average of the same attribute values. Outliers are corrected by the pauta criterion, which considers values greater than three times the standard deviation as outliers. ARIMA's linear prediction is combined with XGBoost's nonlinear prediction to estimate the number of users that will be serviced in future based on the processed data. ARIMA is a prediction model that predicts future values only in the time dimension by investigating the relationship between past values and past errors. The XGBoost nonlinear prediction module is based on the concept of Gradient Boosting, using collections of DT to provide an appropriate prediction. According to the predicted results, UAV BSs are autonomously and dynamically deployed to optimise load balancing. The proposed ARIMA-XGBoost prediction based intelligent deployment model is compared with the randomly deployed model of UAV BSs and the model where UAV BSs are not deployed. The obtained simulation data indicates that the model does enhance the load distribution and provide lower CBR and CDR for the highly loaded macro cell.
- **In 2019 [122]**, an SDN-based load-balanced routing model combining RL and DL was proposed to optimise routing and load balancing policies. The proposed model consists of two main components: policy maker and predictor based on NN. The policy maker draws out a global load balancing strategy through available network information collected periodically. The policy is then estimated by the policy predictor. According to the forecast results, the policy maker updates its information to improve policies. The data obtained from the simulation indicates that this model provides better results compared to the shortest path and Round Robin algorithms in terms of latency and network utilisation.
- **In 2019 [66]**, a DRL-based mobile load balancing algorithm and two-tier mobility load balancing architecture were proposed to handle load balancing problems in ultra-dense networks (UDNs). The upper layer uses the k-Means clustering algorithm to group all BSs according to their historical load levels. The lower layer uses DRL-based load balancing algorithms to optimise intra-cluster load balancing. For each cluster with a controller acting as an agent, the optimal load balancing policy is autonomously learned under the asynchronous parallel learning framework. The DRL action is the control of the CIO value between adjacent cells. The reward signal, on the other hand, is the inverse of the maximum load of the cells since it tries to balance the load distribution by mitigating the worst case. The upper layer adapts to dynamic global flow fluctuations from a macro perspective, while the lower layer adjusts the load distribution within the BS group at a more granular level. Stability is improved with a system control mechanism that works online and learns policies offline. However, the weakness of MDP- based RL approaches is that computational complexity becomes unmanageable in ultra-dense HetNets since all possible system states are tracked by the number of BSs and UEs [123].
- **In 2020 [67]**, an ML-powered load balancing routing scheme was proposed using network state information defined in the form of queue length to train the NN and make route predictions. The architecture considers the use of routers, defined as local and central routers. The local router monitors the incoming status of the packets and selects the next hop for transmission. The central router is used to detect the queue usage and traffic rate of all local routers. Resource utilisation for the next time slot is estimated by deep neural network (DNN) algorithms to fight against network congestion caused by sudden traffic bursts. However, using the resource utilisation criterion alone is insufficient to achieve proper load balancing. Since the connection quality was not considered, the packet loss ratio and delay are high.
- **In 2020 [69]**, a DRL scheme was proposed to solve the load imbalance problem in LTE cellular networks. The proposed model tunes the CIO values of cells as the steering action. A central agent monitors KPIs at the network level and adjusts the CIO value accordingly, preventing cell congestion. The collected simulation data indicates that this algorithm improves the overall throughput compared to the 0 dB CIO algorithm. However, only static UEs were used in the proposed algorithm.
- **In 2020 [68]**, a DRL-based MRO scheme was proposed to learn the optimum parameter values used to describe the mobility patterns of cells. The optimal mobility setting for HO parameters depend on the UE distribution and their velocity. A mobility-sensitive load balancing approach was also offered to configure parameters according to the mobility model of each UE. The performance of the stochastic load balancing scheme was compared to one that implements a long short-term memory for ICIC, mobility load balancing and fixed mobility load balancing approach. The data obtained from the simulation indicates that this method collectively reduces the number of HOF, CBR and unsatisfied users.
- **In 2020 [70]**, an RL-based load balancing scheme was proposed for hybrid WiFi/LiFi networks. RL is trained to

determine the most optimal policy using the trust region policy optimisation. RL is also used to estimate the best access point assignment for a specific situation with a determined optimal policy. The data obtained from the simulation indicates that the RL algorithm provides similar performance to the exhaustive search scheme in a low complexity scenario. It also outperforms the signal strength strategy scheme and the iterative algorithm in most scenarios. However, the effect of receiver orientation or HO overhead were not considered in the system model.

- **In 2020 [71]**, two Q-Learning based cell selection strategies were combined to overcome load and energy imbalances in HetNets. Each UE in the network model has a sensor and a learning processor. The sensing module collects information about the UE from the network. It obtains the power of the signals received from the BSs in the downlink and acquires the remaining energy of the neighbouring UEs in the uplink. The task of the learning processor is to make routing decisions. It must choose the optimal CIO values that provide load balancing in both the downlink and the uplink. It should also choose the best routing destination according to the energy status of the neighbouring BSs.
- **In 2020 [124]**, a congestion control model based on multitasking DRL was proposed in SDN-based networks. In the proposed model, multitask learning is used for tool training. The main task is congestion control, and the auxiliary task is load balancing. The multitasking DRL agent gathers information from the data plane, and the states of two tasks are entered in two separate CNNs. The output layers are then concatenated to create a joint representation of the network state. The environment takes two specific actions from the agent for the two tasks. Actions are distributed over the network to obtain the updated network status, and the rewards of these two agents are computed based on the updated network status. The sum of these two rewards is the overall reward value fed back to the multitasking DRL agent, and the multitasking DRL agent adjusts parameters based on the overall reward.
- **In 2020 [125]**, an intelligent hybrid intra-network load balancing scheme was proposed using multi-agent actor-critical RL to dynamically schedule network traffic. This architecture consists of a central learning and distributed execution framework. The central critic is enforced by the global network state loaded from each switch to streamline the distributed agent training process, thus helping the switches act in a globally coordinated manner. Its performance was compared with the single agent actor-critic RL algorithm and a greedy algorithm. The simulation data indicates that this algorithm has better convergence speed and performance. However, increasing the topological complexity causes the average reward to decrease.
- **In 2020 [72]**, an intelligent SDN architecture using DL was proposed for the vehicle-to-everything (V2X) network. The traffic offloading problem is formulated as a multi-objective optimisation problem. An online-offline approach powered by DL is recommended to solve this optimisation issue. In the online phase, the offload problem is divided into a sub-problem relationship between access points and users, and the relationship between vehicles and latency sensitive users. The Pareto optimal is used to determine solutions to the sub- problems. In the offline phase, DL is used to learn from past optimisation information of the online phase.
- **In 2021 [126]**, an algorithm based on Q-Learning was proposed for managing the HO of UAV BS between macro BSs. The algorithm adjusts the CIO values of cells according to the traffic load status of macro BSs. The load of macro BSs is defined as states, while the reward is defined as the capacity of users served by the UAV-BS. The simulation data indicates that the capacity and UE satisfaction have increased. However, additional complex network scenarios for UAV-BSs were not fully studied.
- **In 2021 [73]**, a DRL-based load balancing model that adjusts the CIO values and transmission powers of cells for load balancing in a homogeneous network was proposed. A central agent takes PRB utilisation, network throughput and users' MCS as the state of the environment, adjusting the CIO and transmission powers of cells according to the network state. The model uses sum throughput as a reward function to better analyse the effect of transmission power on network performance. This model was compared with two other models: one that only adjusts the CIO value and another where the CIO value is zero and the transmission power of cells is constant. The results indicate that this model has similar results in terms of load balancing with the model that only adjusts CIO. The throughput performance did improve by up to 6.5%. The model outperforms the base model in both load balancing and throughput. This paper did not examine how the model will perform in complex scenarios involving different cell types.
- **In 2021 [127]**, two different RL based load balancing models were proposed to adjust the CIO parameter of cells in a homogeneous network. The load balancing techniques are based on Q-Learning and SARSA. These two techniques differ when updating the action-state function during the selection of action in the next step. In both techniques, the state of the environment is the load state of the cells, and the actions tune the CIO value. The proposed techniques were compared with models using a fixed 6 dB CIO and a fixed 12 dB CIO. The results indicate that these two techniques outperform the 6 dB CIO model but not the 12 dB CIO model for balancing the load state of the main cell. Considering the load state of both the main cell and neighbouring cells, the two proposed techniques perform much better

than the models with fixed CIO. The models with fixed CIO only offload the main cell without considering the load state of neighbouring cells. In contrast, RL-based solutions determine CIO parameters by considering the load states of both the main cell and neighbouring cells. However, the limit on the number of states-actions for RL-based solutions increases the computational cost in more complex scenarios.

- **In 2021 [74]**, a clipped double Q-Learning (CDQL) based load balancing algorithm was presented to determine the CIO value of each cell. The proposed algorithm observes the performance parameters and PRB utilisation based on the number of UEs in each cell from the environment, adjusting the CIO values of cells through a central agent. The algorithm was compared with the max-RSRP HO algorithm and the HO algorithm, which considers the utilisation of PRB. Data collected from the simulation indicates that the algorithm improves throughput, latency, jitter and PLR performances compared to baseline algorithms. However, the average PRB utilisation performance of the proposed algorithm may not meet performance targets due to the increase in the number of UEs. In the scenario where UEs are mobile, the algorithm performs similar to the max-RSRP handover algorithm. As stated in the paper, this algorithm is more suitable for application in C-RAN.

IV. RESEARCH CHALLENGES

Although ML algorithms provide great opportunities in solving the load balancing problem, obvious challenges must still be overcome. This section explains the issues present when applying ML to load balancing problems.

A. THE GROWTH OF BIG DATA

To make optimum load balancing decisions, advanced network architectures must support network data provider tools that enable the creation of the big data concept (such as performance data, load state data, configuration data, etc.) provided from the entire network. The big data concept, besides its unique potential, requires new thinking methods and new algorithms in the ML field to make full sense of the large amounts of data produced and to address various challenges.

1) HIGH VOLUME OF BIG DATA

The primary feature of big data that poses a challenge to ML is the ever-expanding volume of network data. Parallel programming methods can be used to reduce the hassle associated with high volumes. Distributed ML approaches can provide a solution to large-scale ML algorithm complexity and memory constraint issues by allocating the learning process to multiple computers or processors. Hybrid approaches that ensure model and data parallelism by simultaneously partitioning both data and model variables allow ML applications to run efficiently in distributed clusters when both the data and the model are too large to fit into a single machine's memory [128]. Cloud computing is a significant advancement that can be used to tackle the large volume problem of big data in future networks. It can be used in HetNets for load balancing since it is capable of increasing computing and storage capacity through the cloud infrastructure, enabling optional resources such as computing power and stored data on remote servers for highly complex ML algorithms that require significant data. Although it collects a large amount of data from operators for subscribers on a daily basis, a large amount of unused data still exists (referred to as dark data) [103]. The need for additional data is evident to create more effective solutions to the load balancing problem, therefore, utilising dark data with optimum potential can increase the efficiency in HetNets.

2) VARIETY OF BIG DATA

Data diversity is another challenge in the big data concept. It describes the structural diversity of the dataset and the types of data it contains, as well as the variety, semantic interpretation and sources of what it represents [129]. Learning from heterogeneous data provided by different network sources is a factor that increases the degree of complexity. An effective solution is to integrate features learned at different levels into the model by studying data representations from each data source [130]. Since big data is a combination of data from various and unknown sources, it is often described as noisy.

3) Veracity of big data

The accuracy and reliability of the source data is also a major challenge in the big data concept since data sources are extremely diverse and data quality cannot be fully verified. Learning from such unproven data, which can also be inaccurate, may cause data misinterpretation and affect the prediction accuracy. The noise caused by the provision of data from various sources may further affect the performance of ML by potentially providing data in an inappropriate way.

4) STORAGE CAPACITY PROBLEM

ML algorithms with good amounts of data require more memory to retain data and use it to train the models. Caching content has been designed for the storage capacity problem in HetNets, especially to eliminate the restriction of limited memory in vehicle networks where behavioural patterns formed by past experiences are frequently used. This has led to reduced latency and the rapid adaptation to radio link conditions, improving the QoE of end users.

5) VELOCITY OF BIG DATA

Velocity of big data refers to the velocity at which data is produced and must be analysed. An ML model must constantly deal with the flow of changing data by rapidly interacting with its environment. In a typical ML model, the system is already trained over the existing training set and the new data entered into the system performs the learned task. However, in some cases, ML cannot automatically learn when new data arrives. The model may become out of date and may not reflect the current state of the system. Online learning is a promising solution to the need for real-time or near-real-time processing of data [129]. Online learning models update according to each new input, providing the ability for data to adapt to new models and react instantly.

B. IMPLEMENTATION OF THE MILLIMETER WAVE (MMWAVE)

Bandwidth bottleneck is the main problem of 5G wireless networks since most available spectrums in microwave frequency are occupied. The mmWave bands are a potential candidate to meet the enormous increase in mobile traffic that will be created by different devices and services added to the 5G/6G systems. These bands can solve the issue of scarce spectrum resources thanks to their large bandwidth range (between 30 to 300 GHz) [131]. However, mmWave signals are sensitive to high path loss, rapidly changing channel conditions and congestion. Therefore, these signals must be propagated over shorter distances compared to existing RF signals. The mmWave cells must be densely deployed throughout the network to meet coverage and capacity requirements of cellular networks [132]. This will lead to a high HO probability in scenarios where both the user and obstacles are mobile. The probability of a high number of HOs can lead to high PPHO and RLF rates, increased outage and decreased throughput [4]. Since the dense deployment of mmWave cells causes significant and rapid load fluctuations, intelligent load balancing models are needed to attain an optimised load distribution. Considering the probabilistic line-of-sight (LOS), non-LOS (NLoS) cases and the irregular nature of mmWave propagation, it is obvious that user association in mmWave cells is different from other systems, thereby requiring new approaches. In [133], a DRL-based load balancing algorithm was proposed for resource allocation and HO management in the mmWave network. The algorithm allocates a spare cell group for the possibility that users will HO in the next time frame based on statistical data and experience. The user association problem was modelled as a non-convex optimisation problem, using the DRL algorithm as an optimisation approach.

C. SECURITY AND PRIVACY

Establishing a unified privacy and security policy will be challenging due to the diversity of vast amounts of data generated by network resources and the inconsistencies in their level of detail. In terms of load balancing, malicious users can perform denial-of-service (DoS) attacks that redirect all traffic payloads to a single data centre. While the resources of this centre are depleted, other resources become unusable [134]. However, an ML-based load balancing system can fool sub-optimal energy consumption or DoS attacks where functions are packaged into several servers with depleted resources. Privacy-sensitive distributed techniques are promising solutions for the secure and efficient processing and storage of personal user data such as location and habits [135]. Applying federated learning algorithms in load balancing models will be useful to ensure the privacy and security of users' personal information [136].

V. FUTURE DIRECTIONS

Integration of advanced ML techniques and next-generation solutions is needed to create more efficient load balancing models. In this section, it examines and discusses the future research directions of ML algorithms, possible new solutions in HetNets, and new technologies that have entered our lives with 5G/6G in terms of load balancing.

A. SDN / NETWORK FUNCTION VIRTUALIZATION (NFV)

SDN technology increases the flexible, dynamic and programmable functionality of network systems by providing a global overview and centralised control. The control and data plane are separated in the network [137]. Thanks to its central control feature, the SDN controller can globally monitor the load levels of nodes in the network. The network traffic can be divided into multiple flow paths through intelligent load balancing techniques in case of congestion in any flow path [20]. The SDN technology effectively allocates network resources to improve network performance and user's QoS. NFV complements SDN network technology and enables the software implementation of network functions to be separated from infrastructure and underlying hardware [8]. The use of such technology allows for on-demand sharing and portability of network resources, thus enabling operators to allocate resources in different locations. NFV and SDN are two potential functions that can be used to create intelligent and flexible services to enable load balancing and efficient resource allocation, especially in cloud.

B. DEEP LEARNING

DL approaches extend the models by adding "depth", i.e., complexity, to classical ML models. Data is hierarchically represented with various levels of abstraction. NNs are made deeper by increasing the number of hidden layers and neurons. They are necessary to process higher dimensional data and learn increasingly complex models. In each layer, neurons train a feature representation based on the output of the previous layer, thanks to the feature hierarchy that enables the management of high-dimensional datasets. Information is hierarchically extracted from raw data from multiple layers of nonlinear processing units to take action or make predictions against specific targets [138]. The main advantage of DL is that it can automatically extract high-level features from complex-structured data and internal correlations without the need for a human- designed learning process [139]. DL handles large amounts of data and provides further benefits since training with big data prevents overfitting of the model. A single model is sufficient to achieve multiple goals in DL. This eliminates the need to repeatedly train the model for different tasks. Although DL possesses unique advantages, it does have several restrictive shortcomings. The approach is full of unknowns and it is not fully understood how certain decisions are made. DL is also vulnerable to attacks that may trigger model mistuning. Collecting large amounts of data to train the DL model from the network can be costly as well. The need for complex DL structures to achieve satisfactory accuracy performance can further cause computational difficulties.

In recent years, new techniques for training big data and deeper networks have created a great opportunity for DL research, along with technological advancements such as the availability of more powerful computers, faster networking and better software infrastructure. DL methods have proven to be effective in integrating data into the models generated from different sources [130]. This indicates that DL techniques are relatively applicable in HetNets. Therefore, load balancing models where DL methods have been integrated with ML techniques have become a solution in recent studies. This prevalence may further increase in future research. Figure 5 shows a DL-based centralized load balancing model in a HetNet. The DL model is fed with data collected from the network and the outputs of this data processed by the DL are applied to the network. For instance, the load states of the cells and the distribution of users can be the input of the DL model, while the output can be the CIO values of the cells [66].

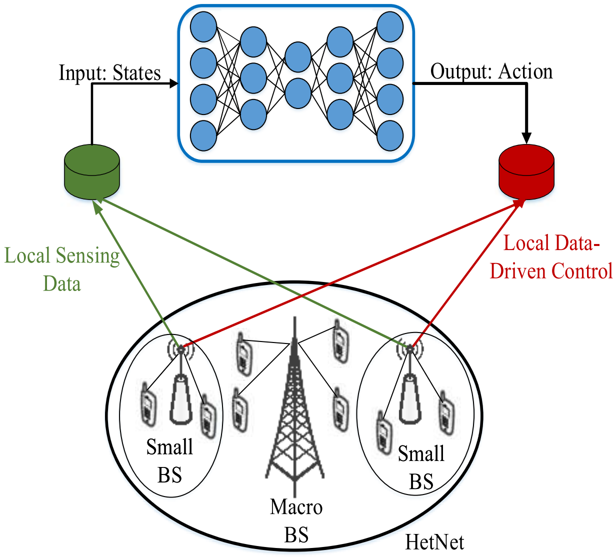

FIGURE 5. Demonstrates a DL-based centralized load balancing model in a HetNet. The DL model is fed with the data collected from the cells and the HO parameters are determined according to the output of the DL model [66].

C. DYNAMIC DEPLOYMENT OF UAV BSS

UAV BSs are an effective solution to coverage problems that arise from traffic fluctuations since they properly adjust the position of UAVs, also known as drones. One of the most challenging and critical aspect is determining the optimal position of UAV BSs to dynamically meet evolving data demands and maximise benefits for operators [140]. Two strategies are present for deploying UAV BSs: deploying before requests arrive and deploying according to real-time demands. In the first strategy, future flow demands are predicted using historical social data. UAV BSs are deployed in advance at points where capacity is thought to exceed in future, increasing the network capacity without congestion in that area. Deploying UAV BSs based on historical data may be inadequate to adapt to dynamic changes that may occur in the network, causing performance fluctuations. Therefore, the second strategy (deploying UAVs according to real-time demands) can be used to increase user experience. By analysing real-time social data, the QoE levels of users are found and the number of UAVs and deployment positions required to maximise QoE can be determined by adopting ML. In Figure 6, UAV BSs deployed at points where traffic bursts are predicted to occur through data analysis prevent overloading of cells. With better understanding, UAV systems can be widely applied to solve the load balancing problem.

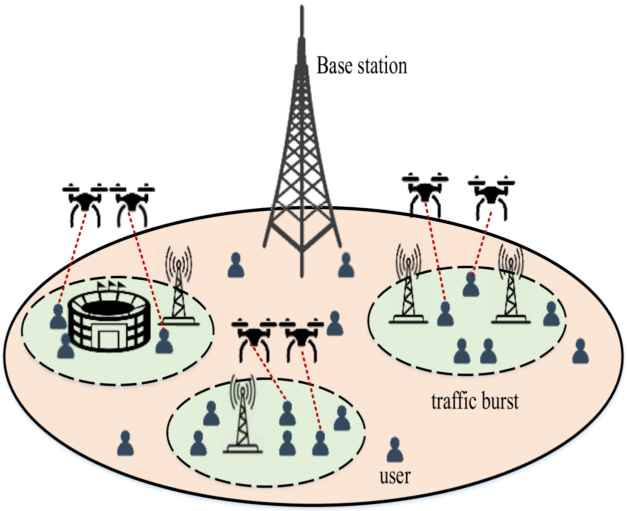

FIGURE 6. With the intelligent distribution of UAV BSs at traffic burst points such as stadiums, overloading of cells is prevented, increasing the service quality and service experience provided to users.

D. TRANSFER LEARNING (TL)

TL utilises and synthesises distilled knowledge from similar tasks and previous experience to facilitate the learning of new problems. It also improves the learning and convergence rate, thereby increasing the robustness of ML methods against different wireless environments while saving energy [141]. TL uses DL features and applies a DNN trained in a different application instead of training NN from scratch. DNNs, with TL included, greatly enhance the training process. At the start of the learning process in DRL, significant time can be spent exploring the environment before achieving an optimal policy. TL techniques can be integrated into DRL to accelerate the learning process [142]. The difficulty in implementing TL is that the source and destination network scenarios may differ. In such cases, it is necessary to carefully design how to extract useful information from historical data when applying TL [143].

TABLE 4. Summary of abbreviations.

Abbreviation	Definition
3GPP	Third generation partnership project
5G	Fifth generation
6G	Sixth generation
ABS	Almost blank subframes
ARIMA	Autoregressive integrated moving average
AI	Artificial intelligence
BBU	Baseband unit
BS	Base station
C-RAN	Cloud radio access network
CAC	Call admission control
CBR	Call blocking ratio
CCO	Concurrent optimization of coverage
CDR	Call dropping ratio
CE	Cross entropy
CIO	Cell individual offset
CDQL	Clipped double Q-Learning
CNN	Convolutional neural network
CRE	Cell range expansion
D2D	Device to device
DT	Decision trees
DL	Deep learning
DNN	Deep neural network
DoS	Denial of service
DRL	Deep reinforcement learning
e-ICIC	Enhanced inter-cell interference coordination
eNB	Evolved NodeB
FL	Fuzzy logic
FLC	Fuzzy logic controller
FQL	Fuzzy Q-Learning
FS	Fuzzy system
GOL	Generic online learning
HetNet	Heterogeneous network
HO	Handover
HOF	Handover failure
HOM	Handover margin
HOR	Handover rate
IoT	Internet of Things
k-NN	k-Nearest Neighbor
KPI	Key performance indicators
LoS	Line-of-sight
LTE	Long term evaluation
MDP	Markov decision process
ML	Machine learning
mmWave	Millimeter wave
MRO	Mobility robustness optimization
NFV	Network functions virtualization
NLP	Nonlinear programming
NN	Neural network
nLOS	Non-line-of-sight
OR	Outage ratio
PCA	Principal component analysis
PLR	Packet loss ratio
PRB	Physical resource block
QoE	Quality of experience
QoS	Quality of service
RL	Reinforcement learning
RLF	Radio link failure
RSRP	Reference signal received power
RSS	Received signal strength
SDN	Software-defined networking
SINR	Signal to interference plus noise ratio
SOM	Self Organizing Maps
SVM	Support vector machines
TL	Transfer learning
TTT	Time to trigger
UAV	Unmanned aerial vehicle
UDN	Ultra-dense network
V2X	Vehicle-to-everything
VM	Virtual machine
XGBoost	Extreme gradient boosting

VI. CONCLUSION

This paper provides readers with a comprehensive analysis of ML-based load balancing algorithms. A detailed review of load balancing is carried out, including the concept, algorithms, addressing, KPI, problems and their control. It provides a complete report of the historical development of ML-based load balancing models in HetNets.

First, this paper provides a basic definition of the concept of load balancing and its objectives. It then presents a basic load balancing model and explains each step to follow. This review paper also explains in detail interesting solutions to tackle the load balancing problem in HetNets such as CRE, data analysis based solutions, cell breathing, fuzzy logic solutions and channel borrowing used in load balancing. The KPIs used to evaluate the performance of load balancing models are introduced with their formulations. In addition, the mathematical modeling of the user association optimization problem is presented with the aim of maximizing the utility function under resource constraints. Also, some clues for realizing optimal association scenarios between users and BSs are shared. The paper also provides a detailed analysis of the working principles of CCO and MRO, which are among the functions of self-organized networks, and the relationship between these two functions with load balancing as well as their joint optimization. The control methods of load balancing algorithms are also presented.

Specifically, the paper provides insights on the applications of ML methods on solving the load balancing problem in HetNets. We begin by presenting the generic ML methods and their distinct categories. Then, we provide a detailed review of technical issues relating to the implementation of load balancing models based on ML technology, analyses of performances, and the shortcomings of the models available in the literature from 2013 to 2021 which are summarized in Table 3. We then describe the research challenges, big data, security, and implementation of mmWave. Finally, we discusses research aspects of future load balancing models such as SDN/NFV, DL, UAV BSs and TL.

APPENDIX

See Table 4.

REFERENCES

[1] K. Addali, "Mobility and resource management for 5G small cell networks," Ph.D. dissertation, École de technologie supèrieure, 2020.

[2] J. Du, C. Jiang, J. Wang, Y. Ren, and M. Debbah, "Machine learning for 6G wireless networks: Carrying forward enhanced bandwidth, massive access, and ultrareliable/low-latency service," *IEEE Veh. Technol. Mag.*, vol. 15, no. 4, pp. 122–134, Dec. 2020.

[3] E. Gures, I. Shayea, A. Alhammadi, M. Ergen, and H. Mohamad, "A comprehensive survey on mobility management in 5G heterogeneous networks: Architectures, challenges and solutions," *IEEE Access*, vol. 8, pp. 195883–195913, 2020.

[4] I. Shayea, M. Ergen, M. H. Azmi, S. A. Colak, R. Nordin, and Y. I. Daradkeh, "Key challenges, drivers and solutions for mobility management in 5G networks: A survey," *IEEE Access*, vol. 8, pp. 172534–172552, 2020.

[5] I. Siomina and D. Yuan, "Load balancing in heterogeneous LTE: Range optimization via cell offset and load-coupling characterization," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 1357–1361.

[6] M. Tayyab, X. Gelabert, and R. Jäntti, "A survey on handover management: From LTE to NR," *IEEE Access*, vol. 7, pp. 118907–118930, 2019.

[7] V. V. Nimmalapudi, A. K. Mengani, R. Vuppula, and R. Jashvantbhai Pandya, "Deep learning based load balancing for improved QoS towards 6G," 2020, *arXiv:2006.16733*.

[8] H. Fourati, R. Maaloul, and L. Chaari, "A survey of 5G network systems: Challenges and machine learning approaches," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 2, pp. 385–431, Feb. 2021.

[9] J. Andrews, S. Singh, Q. Ye, X. Lin, and H. Dhillon, "An overview of load balancing in HetNets: Old myths and open problems," *IEEE Wireless. Commun.*, vol. 21, no. 2, pp. 18–25, Apr. 2014.

[10] S. Mishra and N. Mathur, "Load balancing optimization in LTE/LTE—A cellular networks: A review," 2014, *arXiv:1412.7273*.

[11] M. A. Gadam, M. A. Ahmed, C. K. Ng, N. K. Nordin, A. Sali, and F. Hashim, "Review of adaptive cell selection techniques in LTE-advanced heterogeneous networks," *J. Comput. Netw. Commun.*, vol. 2016, pp. 1–12, 2016.

[12] A. S. Milani and N. J. Navimipour, "Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends," *J. Netw. Comput. Appl.*, vol. 71, pp. 86–98, Aug. 2016.

[13] E. J. Ghomi, A. M. Rahmani, and N. N. Qader, "Load-balancing algorithms in cloud computing: A survey," *J. Netw. Comput. Appl.*, vol. 88, pp. 50–71, Jun. 2017.

[14] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, and A. Rezaee, "Load balancing mechanisms in the software defined networks: A systematic and comprehensive review of the literature," *IEEE Access*, vol. 6, pp. 14159–14178, 2018.

[15] B. Pourghebleh and V. Hayyolalam, "A comprehensive and systematic review of the load balancing mechanisms in the Internet of Things," *Cluster Comput.*, pp. 1–21, 2019.

[16] A. Chandak and N. K. Ray, "A review of load balancing in fog computing," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Dec. 2019, pp. 460–465.

[17] P. Kumar and R. Kumar, "Issues and challenges of load balancing techniques in cloud computing: A survey," *ACM Comput. Surveys*, vol. 51, no. 6, pp. 1–35, Nov. 2019.

[18] M. Mehra, S. Maurya, and N. K. Tiwari, "Network load balancing in software defined network: A survey," *Int. J. Appl. Eng. Res.*, vol. 14, no. 2, pp. 245–253, 2019.

[19] M. R. Belgaum, S. Musa, M. M. Alam, and M. M. Su'ud, "A systematic review of load balancing techniques in software-defined networking," *IEEE Access*, vol. 8, pp. 98612–98636, 2020.

[20] T. Semong, T. Maupong, S. Anokye, K. Kehulakae, S. Dimakatso, G. Boipelo, and S. Sarefo, "Intelligent load balancing techniques in software defined networks: A survey," *Electronics*, vol. 9, no. 7, p. 1091, Jul. 2020.

[21] M. Hamdan, E. Hassan, A. Abdelaziz, A. Elhigazi, B. Mohammed, S. Khan, A. V. Vasilakos, and M. N. Marsono, "A comprehensive survey of load balancing techniques in software-defined network," *J. Netw. Comput. Appl.*, vol. 174, Jan. 2021, Art. no. 102856.

[22] J. Li, L. Ma, Y. Fu, D. Ma, and A. Xiao, "Load balancing in heterogeneous network with SDN: A survey," in *Proc. China Conf. Wireless Sensor Netw.* Springer, 2021, pp. 250–261.

[23] A. Kumar and D. Anand, "Study and analysis of various load balancing techniques for software-defined network (a systematic survey)," in *Proc. Int. Conf. Big Data, Mach. Learn. their Appl.* Springer, 2021, pp. 325–349.

[24] V. Mohammadian, N. J. Navimipour, M. Hosseinzadeh, and A. Darwesh, "Fault-tolerant load balancing in cloud computing: A systematic literature review," *IEEE Access*, vol. 10, pp. 12714–12731, 2022.

[25] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, "Load balancing techniques in cloud computing environment: A review," *J. King Saud Univ. Comput. Inf. Sci.*, Mar. 2021.

[26] A. A. Alkhatib, A. Alsabbagh, R. Maraqa, and S. Alzubi, "Load balancing techniques in cloud computing: Extensive review," *Adv. Sci., Technol. Eng. Syst. J.*, vol. 6, no. 2, pp. 860–870, Apr. 2021.

[27] Z. Ning, Q. Song, L. Guo, M. Dai, and M. Yue, "Dynamic cell range expansion-based interference coordination scheme in next generation wireless networks," *China Commun.*, vol. 11, no. 5, pp. 98–104, May 2014.

[28] M. G. Calabuig, "Self-organizing networks," M.S. thesis, Universitat Politècnica de Catalunya, 2018.

[29] K. Okino, T. Nakayama, C. Yamazaki, H. Sato, and Y. Kusano, "Pico cell range expansion with interference mitigation toward LTE-advanced heterogeneous networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2011, pp. 1–5.

[30] P. Muñoz, "Optimization of mobility parameters using fuzzy logic and reinforcement learning in self-organizing networks," M.S. thesis, Univ. Málaga, 2014.

[31] *Radio Resource Control (RRC)(Release 13)*, Standard TS 36.331, 3GPP.

[32] P. Tian, H. Tian, J. Zhu, L. Chen, and X. She, "An adaptive bias configuration strategy for range extension in LTE-advanced heterogeneous networks," in *Proc. IET Int. Conf. Commun. Technol. Appl. (ICCTA)*, Oct. 2011, pp. 336–340.

[33] M. M. Hasan, S. Kwon, and J.-H. Na, "Adaptive mobility load balancing algorithm for LTE small-cell networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2205–2217, Apr. 2018.

[34] K. Kikuchi and H. Otsuka, "Parameter optimization for adaptive control CRE in HetNet," in *Proc. IEEE 24th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2013, pp. 3334–3338.

[35] E. Rakotomanana and F. Gagnon, "Optimum biasing for cell load balancing under QoS and interference management in HetNets," *IEEE Access*, vol. 4, pp. 5196–5208, 2016.

[36] A. Karandikar, N. Akhtar, and M. Mehta, *Mobility Management in LTE Heterogeneous Networks*. Springer, 2017.

[37] J. A. Ayala-Romero, J. J. Alcaraz, and J. Vales-Alonso, "Data-driven configuration of interference coordination parameters in HetNets," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5174–5187, Jun. 2018.

[38] J. Oh and Y. Han, "Cell selection for range expansion with almost blank subframe in heterogeneous networks," in *Proc. IEEE 23rd Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2012, pp. 653–657.

[39] N. Teerasuttakorn, K. Nuanyai, A. Zamani, A. Schmeink, and S. Chantaraskul, "Study of almost blank subframe configurations for traffic offload in HetNets," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 201–206.

[40] I. Demirci and Ö. Korçak, "Cell breathing algorithms for load balancing in Wi-Fi/cellular heterogeneous networks," *Comput. Netw.*, vol. 134, pp. 140–151, Apr. 2018.

[41] S. Yang, W. Zhang, and X. Zhao, "Virtual cell-breathing based load balancing in downlink LTE—A self-optimizing networks," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2012, pp. 1–6.

[42] S. Navaratnarajah, M. Dianati, and M. A. Imran, "A novel load-balancing scheme for cellular-WLAN heterogeneous systems with a cell-breathing technique," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2094–2105, Sep. 2018.

[43] Q. Ma, H. Yang, H. Zhang, K. Xie, and Z. Wang, "Modeling and analysis of daily driving patterns of taxis in reshuffled ride-hailing service market," *J. Transp. Eng., A, Syst.*, vol. 145, no. 10, Oct. 2019, Art. no. 04019045.

[44] Z. Li, C. Wang, and C.-J. Jiang, "User association for load balancing in vehicular networks: An online reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 2217–2228, Aug. 2017.

[45] F. Xia, J. Wang, X. Kong, Z. Wang, J. Li, and C. Liu, "Exploring human mobility patterns in urban scenarios: A trajectory data perspective," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 142–149, Mar. 2018.

[46] G. Cardone, A. Corradi, L. Foschini, and R. Ianniello, "ParticipAct: A large-scale crowdsensing platform," *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 1, pp. 21–32, Jan. 2016.

[47] 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.

[48] J. Hu, H. Zhang, Y. Liu, X. Li, and H. Ji, "An intelligent UAV deployment scheme for load balance in small cell networks using machine learning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.

[49] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller. I," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 404–418, Mar./Apr. 1990.

[50] M. D. Smith, A. R. Bonde, Jr., G. J. Catherine, R. Napolitano, and D. M. Wolfe, "Fuzzy logic-based evidence fusion tool for network analysis," U. S. Patent 5 787 235, Jul. 28, 1998.

[51] G. Feng, *Analysis and Synthesis of Fuzzy Control Systems: A Modelbased Approach*, vol. 37. Boca Raton, FL, USA: CRC Press, 2018.

[52] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller. I," *IEEE Trans. Syst., Man and*, vol. 20, no. 2, pp. 404–418, Mar. 1990.

[53] A. S. Omar, M. Waweru, and R. Rimiru, "A literature survey: Fuzzy logic and qualitative performance evaluation of supply chain management," *Int. J. Eng. Sci.*, vol. 4, no. 5, pp. 56–63, 2015.

[54] J. M. Ruiz-Avilés, S. Luna-Ramírez, M. Toril, and F. Ruiz, "Traffic steering by self-tuning controllers in enterprise LTE femtocells," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, pp. 1–16, Dec. 2012.

[55] J. Rodriguez, I. De 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)*, May 2011, pp. 1–5.

[56] 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)*, May 2011, pp. 1–5.

[57] P. Muñoz, R. Barco, and I. de la Bandera, "Load balancing and handover joint optimization in LTE networks using fuzzy logic and reinforcement learning," *Comput. Netw.*, vol. 76, pp. 112–125, Jan. 2015.

[58] 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, May 2013.

[59] H. Zhang, L. Song, and Y. J. Zhang, "Load balancing for 5G ultra-dense networks using device-to-device communications," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4039–4050, Jun. 2018.

[60] T. Kudo and T. Ohtsuki, "Cell range expansion using distributed Q-learning in heterogeneous networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2013, no. 1, p. 61, 2013.

[61] J. Xu, L. Tang, Q. Chen, and L. Yi, "Study on based reinforcement Q-learning for mobile load balancing techniques in LTE—A HetNets," in *Proc. IEEE 17th Int. Conf. Comput. Sci. Eng.*, Dec. 2014, pp. 1766–1771.

[62] 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)*, Nov. 2015, pp. 1–5.

[63] M. S. Ali, P. Coucheney, and M. Coupechoux, "Load balancing in heterogeneous networks based on distributed learning in near-potential games," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 5046–5059, Jul. 2016.

[64] S. Musleh, M. Ismail, and R. Nordin, "Load balancing models based on reinforcement learning for self-optimized macro-femto LTE-advanced heterogeneous network," *J. Telecommun., Electron. Comput. Eng. (JTEC)*, vol. 9, no. 1, pp. 47–54, 2017.

[65] A. Roshdy, A. Gaber, F. Hantera, and M. ElSebai, "Mobility load balancing using machine learning with case study in live network," in *Proc. Int. Conf. Innov. Trends Comput. Eng. (ITCE)*, Feb. 2018, pp. 145–150.

[66] Y. Xu, W. Xu, Z. Wang, J. Lin, and S. Cui, "Load balancing for ultradense networks: A deep reinforcement learning-based approach," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9399–9412, Dec. 2019.

[67] H. Yao, X. Yuan, P. Zhang, J. Wang, C. Jiang, and M. Guizani, "Machine learning aided load balance routing scheme considering queue utilization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7987–7999, Aug. 2019.

[68] A. Mohajer, M. Bavaghar, and H. Farrokhi, "Mobility-aware load balancing for reliable self-organization networks: Multi-agent deep reinforcement learning," *Rel. Eng. Syst. Saf.*, vol. 202, Oct. 2020, Art. no. 107056.

[69] K. Attiah, K. Banawan, A. Gaber, A. Elezabi, K. Seddik, Y. Gadallah, and K. Abdullah, "Load balancing in cellular networks: A reinforcement learning approach," in *Proc. IEEE 17th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2020, pp. 1–6.

[70] R. Ahmad, M. D. Soltani, M. Safari, A. Srivastava, and A. Das, "Reinforcement learning based load balancing for hybrid LiFi WiFi networks," *IEEE Access*, vol. 8, pp. 132273–132284, 2020.

[71] C. Jialing, Y. Mingxi, D. Xiaohui, and J. Bingli, "Q-learning based selection strategies for load balance and energy balance in heterogeneous networks," in *Proc. 5th Int. Conf. Comput. Commun. Syst. (ICCCS)*, May 2020, pp. 728–732.

[72] B. Fan, Z. He, Y. Wu, J. He, Y. Chen, and L. Jiang, "Deep learning empowered traffic offloading in intelligent software defined cellular V2X networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13328–13340, Nov. 2020.

[73] G. Alsuhli, H. A. Ismail, K. Alansary, M. Rumman, M. Mohamed, and K. G. Seddik, "Deep reinforcement learning-based CIO and energy control for LTE mobility load balancing," in *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2021, pp. 1–6.

[74] P. E. I. Rivera and M. Erol-Kantarci, "QoS-aware load balancing in wireless networks using clipped double Q-Learning," 2021, *arXiv:2107.04207*.

[75] Y. L. Lee, T. C. Chuah, A. A. El-Saleh, and J. Loo, "User association for Backhaul load balancing with quality of service provisioning for heterogeneous networks," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2338–2341, Nov. 2018.

[76] A. J. Fehske, I. Viering, J. Voigt, E. Sartori, S. Redana, and G. P. Fettweis, "Small-cell self-organizing wireless networks," *Proc. IEEE*, vol. 102, no. 3, pp. 334–350, Mar. 2014.

[77] S. A. Khan, A. Kavak, S. Aldirmaz Colak, and K. Kucuk, "A novel fractional frequency reuse scheme for interference management in LTE—A HetNets," *IEEE Access*, vol. 7, pp. 109662–109672, 2019.

[78] M. Hoppari, M. Uitto, J. Mäkelä, I. Harjula, and S. Rantala, "Performance of the 5th generation indoor wireless technologies-empirical study," *Future Internet*, vol. 13, no. 7, p. 180, Jul. 2021.

[79] K. Abdullah, N. Korany, A. Khalafallah, A. Saeed, and A. Gaber, "Characterizing the effects of rapid LTE deployment: A data-driven analysis," in *Proc. Netw. Traffic Meas. Anal. Conf. (TMA)*, Jun. 2019, pp. 97–104.

[80] M. Manalastas, "Realistic modeling of handover events in a multi-carrier 5G network: A preliminary step towards COP-KPI relationship realization," Tech. Rep., 2020.

[81] Z. Ren, P. Fertl, Q. Liao, F. Penna, and S. Stanczak, "Street-specific handover optimization for vehicular terminals in future cellular networks," in *Proc. IEEE 77th Veh. Technol. Conf. (VTC Spring)*, Jun. 2013, pp. 1–5.

[82] C.-C. Lin, K. Sandrasegaran, X. Zhu, and Z. Xu, "On the performance of capacity integrated CoMP handover algorithm in LTE-advanced," in *Proc. 18th Asia–Pacific Conf. Commun. (APCC)*, Oct. 2012, pp. 871–876.

[83] C.-C. Lin, K. Sandrasegaran, and S. Reeves, "Handover algorithm with joint processing in LTE-advanced," in *Proc. 9th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol.*, May 2012, pp. 1–4.

[84] D. Castro-Hernandez and R. Paranjape, "Dynamic analysis of load balancing algorithms in LTE/LTE—A HetNets," *Wireless Pers. Commun.*, vol. 96, no. 3, pp. 3297–3315, Oct. 2017.

[85] D. Liu, L. Wang, Y. Chen, M. Elkashlan, K.-K. Wong, R. Schober, and L. Hanzo, "User association in 5G networks: A survey and an outlook," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1018–1044, 2nd Quart. 2016.

[86] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.

[87] T. Zhou, Z. Liu, J. Zhao, C. Li, and L. Yang, "Joint user association and power control for load balancing in downlink heterogeneous cellular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2582–2593, Mar. 2018.

[88] X. Huang, W. Xu, G. Xie, S. Jin, and X. You, "Learning oriented cross-entropy approach to user association in load-balanced HetNet," *IEEE Wireless Commun. Lett.*, vol. 7, no. 6, pp. 1014–1017, Dec. 2018.

[89] K. Toda, T. Yamamoto, T. Ohseki, and S. Konishi, "Load balancing techniques based on antenna tilt and handover timing control," in *Proc. IEEE 78th Veh. Technol. Conf. (VTC Fall)*, Sep. 2013, pp. 1–6.

[90] A. Asghar, H. Farooq, and A. Imran, "Concurrent optimization of coverage, capacity, and load balance in HetNets through soft and hard cell association parameters," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8781–8795, Sep. 2018.

[91] A. J. Fehske, H. Klessig, J. Voigt, and G. P. Fettweis, "Concurrent load-aware adjustment of user association and antenna tilts in self-organizing radio networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 5, pp. 1974–1988, Jun. 2013.

[92] Z. Liu, P. Hong, K. Xue, and M. Peng, "Conflict avoidance between mobility robustness optimization and mobility load balancing," in *Proc. IEEE Global Telecommun. Conf. GLOBECOM*, Dec. 2010, pp. 1–5.

[93] M. T. Nguyen, S. Kwon, and H. Kim, "Mobility robustness optimization for handover failure reduction in LTE small-cell networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4672–4676, May 2018.

[94] H. Fourati, R. Maaloul, and L. Chaari, "A survey of 5G network systems: Challenges and machine learning approaches," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 2, pp. 1–47, 2020.

[95] A. El Amine, "Radio resource allocation in 5G cellular networks powered by the smart grid and renewable energies," Ph.D. dissertation, Ecole nationale supérieure Mines-Télécom Atlantique Bretagne Pays de la Loire, 2019.

[96] L. Fahrmeir, T. Kneib, S. Lang, and B. Marx, *Regression*. Springer, 2007.

[97] W. S. Noble, "What is a support vector machine?" *Nature Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, 2006.

[98] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, vol. 1, no. 10. New York, NY, USA: Springer, 2001.

[99] C. Kingsford and S. L. Salzberg, "What are decision trees?" *Nature Biotechnol.*, vol. 26, no. 9, pp. 1011–1013, Sep. 2008.

[100] W.-Y. Loh, "Classification and regression trees," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 14–23, Jan. 2011.

[101] S. Haykin and N. Network, "A comprehensive foundation," *Neural Netw.*, vol. 2, p. 41, Feb. 2004.

[102] D. J. Livingstone, *Artificial Neural Networks: Methods and Applications*. Springer, 2008.

[103] 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 Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2392–2431, 4th Quart., 2017.

[104] L. Jiang, Z. Cai, D. Wang, and S. Jiang, "Survey of improving K-nearest-neighbor for classification," in *Proc. 4th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, 2007, pp. 679–683.

[105] S. Sun and Q. Chen, "Distance metric learning for large margin nearest neighbor classification," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 25, no. 7, pp. 1073–1087, Nov. 2011.

[106] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[107] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient K-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.

[108] T. Kohonen, "Essentials of the self-organizing map," *Neural Netw.*, vol. 37, pp. 52–65, Jan. 2013.

[109] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[110] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[111] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette, "Evolutionary algorithms for reinforcement learning," *J. Artif. Intell. Res.*, vol. 11, pp. 241–276, Sep. 1999.

[112] 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, Jan. 1996.

[113] M. Wiering and M. Van Otterlo, *Reinforcement Learning*, vol. 12. Springer, 2012.

[114] L.-A. Phan, D.-T. Nguyen, M. Lee, D.-H. Park, and T. Kim, "Dynamic fog-to-fog offloading in SDN-based fog computing systems," *Future Gener. Comput. Syst.*, vol. 117, pp. 486–497, Apr. 2021.

[115] M. Simsek, M. Bennis, and I. Guvenc, "Mobility management in HetNets: A learning-based perspective," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, pp. 1–13, 2015.

[116] P. Semov, P. Koleva, K. Tonchev, V. Poulkov, and A. Mihovska, "Autonomous learning model for achieving multi cell load balancing capabilities in HetNet," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, 2016, pp. 1–5.

[117] B. Shahriari, M. Moh, and T.-S. Moh, "Generic online learning for partial visible dynamic environment with delayed feedback: Online learning for 5G C-RAN load-balancer," in *Proc. Int. Conf. High Perform. Comput. Simul. (HPCS)*, Jul. 2017, pp. 176–185.

[118] C. Gomez, A. Shami, and X. Wang, "Machine learning aided scheme for load balancing in dense IoT networks," *Sensors*, vol. 18, no. 11, p. 3779, Nov. 2018.

[119] J. Li, G. Luo, N. Cheng, Q. Yuan, Z. Wu, S. Gao, and Z. Liu, "An end-to-end load balancer based on deep learning for vehicular network traffic control," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 953–966, Feb. 2019.

[120] P. H. Barros, I. Cardoso-Pereira, L. Foschini, A. Corradi, and H. S. Ramos, "Load balancing in D2D networks using reinforcement learning," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2019, pp. 1–6.

[121] J.-Y. Baek, G. Kaddoum, S. Garg, K. Kaur, and V. Gravel, "Managing fog networks using reinforcement learning based load balancing algorithm," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–7.

[122] T. T. Huong, N. D. D. Khoa, N. X. Dung, and N. H. Thanh, "A global multipath load-balanced routing algorithm based on reinforcement learning in SDN," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2019, pp. 1336–1341.

[123] N. Raharya, W. Hardjawana, O. Al-Khatib, and B. Vucetic, "Pursuit learning-based joint pilot allocation and multi-base station association in a distributed massive MIMO network," *IEEE Access*, vol. 8, pp. 58898–58911, 2020.

[124] K. Lei, Y. Liang, and W. Li, "Congestion control in SDN-based networks via multi-task deep reinforcement learning," *IEEE Netw.*, vol. 34, no. 4, pp. 28–34, Jul. 2020.

[125] T. Mai, H. Yao, Z. Xiong, S. Guo, and D. T. Niyato, "Multi-agent actor-critic reinforcement learning based in-network load balance," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.

[126] A. Madelkhanova and Z. Becvar, "Optimization of cell individual offset for handover of flying base station," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Apr. 2021, pp. 1–7.

[127] M. Z. Asghari, M. Ozturk, and J. Hamalainen, "Reinforcement learning based mobility load balancing with the cell individual offset," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Apr. 2021, pp. 1–5.

[128] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, "Machine learning on big data: Opportunities and challenges," *Neurocomputing*, vol. 237, pp. 350–361, May 2017.

[129] A. L'Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, "Machine learning with big data: Challenges and approaches," *IEEE Access*, vol. 5, pp. 7776–7797, 2017.

[130] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP J. Adv. Signal Process.*, vol. 2016, no. 1, p. 67, Dec. 2016.

[131] E. Hossain, M. Rasti, H. Tabassum, and A. Abdelnasser, "Evolution toward 5G multi-tier cellular wireless networks: An interference management perspective," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 118–127, Jun. 2015.

[132] A. Alizadeh and M. Vu, "Load balancing user association in millimeter wave MIMO networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 2932–2945, Jun. 2019.

[133] S. Khosravi, H. S. Ghadikolaei, and M. Petrova, "Learning-based load balancing handover in mobile millimeter wave networks," 2020, *arXiv:2011.01420*.

[134] J. Suomalainen, A. Juhola, S. Shahabuddin, A. Mammela, and I. Ahmad, "Machine learning threatens 5G security," *IEEE Access*, vol. 8, pp. 190822–190842, 2020.

[135] N. Zhang, P. Yang, J. Ren, D. Chen, Y. Li, and X. Shen, "Synergy of big data and 5G wireless networks: Opportunities, approaches, and challenges," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 12–18, Feb. 2018.

[136] M. S. Mollel, A. I. Abubakar, M. Ozturk, S. F. Kaijage, M. Kisangiri, S. Hussain, M. A. Imran, and Q. H. Abbasi, "A survey of machine learning applications to handover management in 5G and beyond," *IEEE Access*, vol. 9, pp. 45770–45802, 2021.

[137] M. A. Tunc, E. Gures, and I. Shayea, "A survey on IoT smart healthcare: Emerging technologies, applications, challenges, and future trends," 2021, *arXiv:2109.02042*.

[138] S. Zhang and D. Zhu, "Towards artificial intelligence enabled 6G: State of the art, challenges, and opportunities," *Comput. Netw.*, vol. 183, Dec. 2020, Art. no. 107556.

[139] C. Zhang, "Deep neural mobile networking," 2020, *arXiv:2011.05267*.

[140] Y. Liu, S. Bi, Z. Shi, and L. Hanzo, "When machine learning meets big data: A wireless communication perspective," *IEEE Veh. Technol. Mag.*, vol. 15, no. 1, pp. 63–72, Mar. 2020.

[141] M. Wang, Y. Lin, Q. Tian, and G. Si, "Transfer learning promotes 6G wireless communications: Recent advances and future challenges," *IEEE Trans. Rel.*, vol. 70, no. 2, pp. 790–807, Jun. 2021.

[142] C. T. Nguyen, N. Van Huynh, N. H. Chu, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, Q.-V. Pham, D. Niyato, E. Dutkiewicz, and W.-J. Hwang, "Transfer learning for future wireless networks: A comprehensive survey," 2021, *arXiv:2102.07572*.

[143] W. Wang, J. Zhang, and Q. Zhang, "Transfer learning based diagnosis for configuration troubleshooting in self-organizing femtocell networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–5.

EMRE GURES received the B.Sc. degree in electrical and electronic engineering from Istanbul University, Istanbul, Turkey, in 2017, and the M.Sc. degree in telecommunication engineering from Istanbul Technical University, Istanbul, in 2020, where he is currently pursuing the Ph.D. degree, under the supervision of Dr. Ibraheem Shayea. His current research interests include in mobility management, handover, and load balancing.

IBRAHEEM SHAYEA received the bachelor's degree in electronics and communication engineering from the Faculty of Engineering, University of Diyala, in 2004, and the master's degree in communication and computer engineering and the Ph.D. degree in electrical and electronic engineering, specifically in wireless communication systems, from the Department of Electrical and Electronics Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia (UKM), Malaysia, in 2010 and 2015, respectively. From January 2005 to June 2006, he worked as a computer and electronic maintenance engineer in Yemen. From July 2006 to February 2007, he worked as a Maintenance Manager at the Computer and Research Center (CRC), Yemen. From January 2011 to December 2015 (during his Ph.D. study), he worked as a Research Assistant and a Demonstrator at the Department of Electrical and Electronics Engineering, Faculty of Engineering and Built Environment, UKM. From January 2016 to June 2018, he worked as a Postdoctoral Fellow at the Wireless Communication Center (WCC), University of Technology Malaysia (UTM), Malaysia. From September 2018 to August 2019, he worked as a Researcher Fellow at Istanbul Technical University (ITU), Istanbul, Turkey. Since September 2019, he has been working as an Associate Researcher with the Department of Electronics and Communication Engineering, Faculty of Electrical and Electronics Engineering, ITU. He is teaching courses for bachelor's and master's students. He has published several scientific research journals, conference papers, and whitepapers. He is running research projects with postdoctoral researchers, Ph.D., and master's students. His research interests include mobility management in future heterogeneous (4G, 5G, and 6G) networks, mobile edge computing, machine and deep learning, the Internet of Things (IoT), propagation of millimetre-wave, mobile broadband technology, future data traffic growth, and spectrum gap analysis.

MUSTAFA ERGEN received the B.S. degree in electrical engineering as valedictorian from ODTU, the M.A. degree from International Studies, the M.S. and Ph.D. degrees in electrical engineering from UC Berkeley, and the MOT Program from the HAAS School of Business. He is currently a Professor at Istanbul Technical University. Previously, he was at Turk Telekom as a Chief Technology Advisor. Earlier, he co-founded Silicon Valley startup WiChorus Inc., focusing 4G technologies and company is acquired by Tellabs [now Coriant]. He was also a National Semiconductor Fellow [now TI] at the University of California Berkeley, where he co-founded the Distributed Sensing Laboratory, focusing on statistical sensor intelligence and vehicular communication. He is the Founder of Venture funded Ambeent Inc., focusing 5G WiFi and AI. He has more than 50 patent applications, many publications, and authored three books *Mobile Broadband* and *Multi-Carrier Digital Communications* (Springer) are about wireless communications, and the entrepreneurship *Entrepreneurial Capital: Silicon Valley History and Startup Economy* (Turkish: Alfa).

MARWAN HADRI AZMI received the B.Eng. degree (Hons.) in electrical and telecommunications from Universiti Teknologi Malaysia, in 2003, the M.Sc. degree in communications and signal processing from the Imperial College of Science, Technology and Medicine, University of London, in 2005, and the Ph.D. degree from the University of New South Wales, Australia, in 2012. From 2012 to 2014, he spent his Sabbatical leave of absence at the McGill University, Canada, working in the funded project by RIM Inc. and NSERC, entitled "Cooperative Spectrum Sensing and Information Relaying in Cognitive Wireless Communications." He is currently a Senior Lecturer at the Wireless Communication Centre, Universiti Teknologi Malaysia. His research interests include mobile and wireless communications, communication theory, error control coding, relay networks, spectrum sensing for cognitive radio, and iterative receiver.

AYMAN A. EL-SALEH (Senior Member, IEEE) received the B.Sc. degree in communications engineering from Omar Al-Mukhtar University (OMU), Libya, in 1999, and the M.Sc. degree in microelectronics engineering and the Ph.D. degree in wireless communications from Universiti Kebangsaan Malaysia (UKM), in 2006 and 2012, respectively. In October 2006, he joined the Faculty of Engineering, Multimedia University (MMU), as a Senior Lecturer and the Chairperson of the Centre for Wireless Technology (CWT). In April 2017, he was appointed as the Lead of Connectivity Research Theme at MMU. In September 2017, he joined A'Sharqiyah University, Oman, where he is currently an Assistant Professor. His research interests include cognitive radio networks, heterogeneous LTE/LTE-advanced cellular networks, and applications of artificial intelligence and evolutionary algorithms in wireless communications. He was an Executive Committee Member of the IEEE Malaysia ComSoc/VTS Joint Chapter, in 2015 and 2016. He is a reviewer of several ISI-indexed journals and published about 60 journals and conference papers.