A prediction-model-assisted reinforcement learning algorithm for handover decision-making in hybrid LiFi and WiFi networks

DAYRENE FROMETA FONSECA[1,2,*], BORJA GENOVES GUZMAN[3], GIOVANNI LUCA MARTENA[4], RUI BIAN[4], HARALD HAAS[4], AND DOMENICO GIUSTINIANO[1]

[1] IMDEA Networks Institute, Av. Mar Mediterraneo 22, Leganes, Madrid, 28918, Spain
[2] University Carlos III of Madrid, Av. de la Universidad, 30, Leganes, Madrid, 28911, Spain
[3] University of Virginia, North Emmet Street 1001, Charlottesville, VA 22903, USA
[4] PureLiFi Ltd., 51 Timber Bush, Edinburgh, Scotland, EH6 6QH, UK
[*] Corresponding author: dayrene.frometa@imdea.org

Compiled December 6, 2023

The handover process in hybrid light fidelity (LiFi) and wireless fidelity (WiFi) networks (HLWNets) is very challenging due to the short area covered by LiFi access points and the coverage overlap between LiFi and WiFi networks, which introduce frequent horizontal and vertical handovers, respectively. Different handover schemes have been proposed to reduce the handover rate in HLWNets, among which handover skipping (HS) techniques stand out. However, existing solutions are still inefficient or require knowledge that is not available in practice, such as the exact user's trajectory or the network topology. In this work, a novel machine learning-based handover scheme is proposed to overcome the limitations of previous HS works. Specifically, we have designed a classification model to predict the type of user's trajectory and assist a reinforcement learning (RL) algorithm to make handover decisions that are automatically adapted to new network conditions. The proposed scheme is called RL-HO, and we compare its performance against the standard handover scheme of long-term evolution (STD-LTE) and the so-called smart handover (Smart HO) algorithm. We show that our proposed RL-HO scheme improves the network throughput by 146% and 59% compared to STD-LTE and Smart HO, respectively. We make our simulator code publicly available to the research community.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

Light Fidelity (LiFi) has been demonstrated to be one of the most promising technologies to alleviate the radiofrequency (RF) spectrum crunch. By exploiting visible wavelengths, LiFi leverages the already deployed light-emitting diode (LED) based illumination infrastructure to provide communication data. Then, LiFi is an optical wireless and free space optics communications technology addressed to short-range scenarios. A recent work studied the techno-economic details of LiFi, demonstrating that providing the existing lighting infrastructure with communication capabilities is very cost-effective [1]. LiFi presents considerable advantages such as better security and, as highlighted, the affordability of its off-the-shelf network elements. However, LiFi signals will easily suffer from light-path blockages in indoor scenarios with many moving elements, resulting in an unstable quality of service. Prior works considered hybrid optical wireless and RF links to reduce the system outage probability [2]. In the case of LiFi, exploiting the massive penetration of Wireless Fidelity (WiFi) with hybrid LiFi and WiFi networks (HLWNets) becomes a promising solution for future indoor wireless communication frameworks. Indeed, Cisco has recently demonstrated that light-based and radio-based communications can coexist using standard-based roaming such as IEEE 802.11r, which will result in an easy integration, deployment, and use of LiFi [3].

WiFi networks usually cover areas with diameters of up to 20 m, whereas a LiFi atto-cell diameter is around 2-3 m [4]. This leads to one of the biggest challenges that HLWNets need to solve: implementing efficient handover (HO) mechanisms for a network where the coverage of two different technologies overlaps. Note that in such a hybrid scenario, using the handover (HO) mechanisms intended for conventional mobile networks (i.e., based on signal strength strategies) will generate a large number of HOs and, thus, an increased outage probability, a large latency, and extra traffic produced in the backhaul.

In a common area with a single WiFi router and multiple LiFi

access points (APs), we can find two types of handovers: 1) horizontal handovers (HHOs), which are produced when the user switches from a LiFi AP to another; and 2) vertical handovers (VHOs), which involves switching to a different technology, from LiFi to WiFi or vice-versa. A HHO is fast and simple because it does not require inter-technology switches for the access network. Multiple HHO techniques have been proposed in the LiFi literature [5–8]. Differently, a VHO involves a longer delay, processing, and network overhead, but it allows for strengthening the transmitter-receiver link with a different communication technology.

In an HLWNet, a LiFi user that is moving fast will produce many HHOs due to the small coverage of individual atto-cells that constitute the whole LiFi network. In such a scenario, switching the fast-moving user to WiFi will reduce the number of handovers and guarantee a wireless connection when the LiFi channel may be blocked, at the expense of a potential throughput decrease. Likewise, when the number of users connected to WiFi is high, and the throughput of each WiFi user is low due to the resource sharing, performing a VHO to switch slow-moving users to LiFi will guarantee a better quality of service for those users, while offloading the WiFi network. A potential approach to reduce the number of VHOs in HLWNets is aggregating LiFi and WiFi capacities with a protocol such as the Multipath Transmission Control Protocol (MPTCP), in which the Internet Engineering Task Force (IETF) has put much effort since 2013 [9]. By including a subflow sequence number within the packet overhead, MPTCP enables a user to be served by LiFi and WiFi simultaneously. However, VHOs in HLWNets are still needed for traffic offloading purposes.

Different works can be found in the literature addressing the handover decision problem in HLWNets. A novel approach was introduced in [10], where the authors exploit the statistical information of channel blockages to propose a centralized handover scheme that employs optimization techniques to maximize the overall system throughput over a period of time. However, because of the high computational complexity, these optimization-based systems are still far from a real implementation. Fuzzy logic [11] and artificial neural networks (ANNs) [12] have been proposed as a potential alternative to the complex problem of handover decision-making in dense hybrid networks, exploiting system parameters such as the channel state information, user speed, required data rate or the network topology. However, fuzzy-logic rules are pre-defined and must be adjusted every time the network changes, whereas the ANN has a high complexity, as claimed by authors in [12]. Other handover techniques in HLWNets are based on a Markov decision process [13], a decision flowchart [14], or evolutionary game theory [15]. However, none of these works consider practical scenarios with mobile users.

Unlike current state-of-the-art approaches, in this paper, we propose a simple and practical HO decision-making algorithm for HWLNets. It is based on reinforcement learning (RL) and dynamically adapts the handover decisions to new network conditions. The proposed RL algorithm is supported by a classification model that predicts the type of user trajectory by exploiting information available at the user equipment, eliminating the need for previous knowledge to carry out the handover decisions.

The remainder of this paper is organized as follows. Section 2 summarizes the work related to the handover schemes presented in the literature. Section 3 introduces the system model, including the corresponding models for the channels, the user mobility, and the line-of-sight path blockages. In Sections 4 and 5 we detail our proposal and the simulation results, respectively. Finally, we draw our conclusion in Section 6.

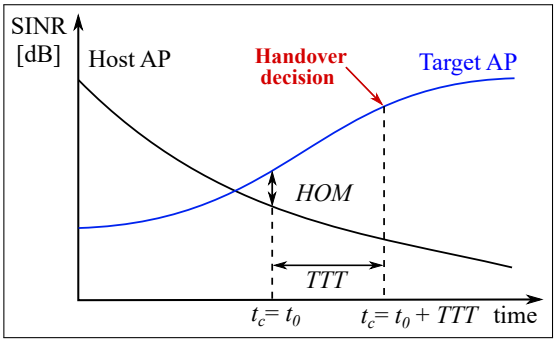

Fig. 1. The hysteresis principle of the standard handover scheme in LTE (STD-LTE).

2. RELATED WORK

In this section, we introduce the handover techniques used in current standards and proposed in the literature that lay the foundation of our proposal.

A. Standard Handover Scheme in LTE

The standard handover scheme in LTE, referred here as STD-LTE, avoids successive handovers by applying hysteresis, which postpones the handover decision for a pre-defined time. Fig. 1 represents the hysteresis principle of STD-LTE. As shown, it is based on two parameters: the handover margin (HOM) and the time to trigger (TTT). Specifically, the handover process is triggered when the signal-to-interference-plus-noise ratio (SINR)[1] of the host AP is *HOM* dBs below the SINR of other AP, here called target AP. However, as depicted in Fig. 1, the handover decision is not immediate. It must wait for a time period equal to *TTT*. That is, it must ensure that the target AP keeps providing the highest SINR for a certain amount of time *TTT*, with a typical value in the order of hundreds of milliseconds [16]. The time counter t_c is running as long as the target AP provides better SINR than the host AP and is reset otherwise. As shown in Fig. 1, when t_c reaches $t_0 + TTT$, a handover decision is made to transfer the user from the host AP to the target AP.

The main motivation for the hysteresis principle of STD-LTE is to avoid the so-called ping-pong effect [16], i.e., it postpones the HO decision to some extent, then avoiding frequent HOs. However, it has limited effectiveness in reducing the number of handovers in ultra-dense networks as the HLWNets. Besides, it has the drawback that in case of LiFi-link blockage, the user will stay disconnected for up to a time given by the *TTT* parameter until a new handover decision is made.

B. Handover Skipping in HLWNets

Previous works in the literature have reduced the number of handovers in HLWNets by skipping the unnecessary ones [17, 18]. Specifically, they exploit SINR measurements to determine whether the user is moving toward the central area of an AP or

[1] In this paper, we use the SINR instead of the reference signal received power (RSRP) to suit the HLWNet. According to the LTE standard, both SINR or RSRP could be used [16]. Note that the target AP is also considered as interference for the SINR computation, as the HO has not been established yet.

along its edge area, skipping those APs that the user crosses by the border of the cell.

Existing HO skipping schemes count time in the same way as STD-LTE and use its hysteresis principle to trigger the HO decisions. Several proceeding methods can be carried out when the *TTT* duration is reached. Instead of immediately switching to the target AP, an objective function can determine a new target AP, which becomes the new host AP as soon as a figure of merit (e.g. SINR) is larger than the one of the current host AP. The handover skipping algorithm in [18], referred here as Smart HO, introduces the following objective function:

$$\Gamma = \begin{cases} \gamma_i^{(t_0)} + \Delta\gamma_i, & \text{for LiFi} \\ \lambda(\gamma_i^{(t_0)} + \Delta\gamma_i), & \text{for WiFi} \end{cases} \quad \textbf{(1)}$$

where $\gamma_i^{(t_0)}$ is the SINR in time instant t_0, which is the triggering time of the handover algorithm, $\Delta\gamma_i$ is the SINR difference between the starting and ending points of the t_c counter (defined in Eq. (6) in [18]), and $\lambda \in (1, \infty)$ is a weight coefficient whose value is optimized according to the user's speed. For fast-moving users, λ should be relatively large to give the selection preference to WiFi and to reduce the number of handovers. For users that are moving slowly, it should be close to 1 to let them access the AP offering the highest SINR.

A more recent work has integrated the user's satisfaction degree into the objective function and it has introduced a dynamic weight coefficient that is trained by an ANN to fit different network scenarios [12]. Although this work has been shown to significantly reduce the number of handovers in HWLNets, it has the limitation of relying on fixed HO decision-making rules, which prevents it from adapting to changing network conditions (i.e., the presence of new blocking elements, new interference sources, etc.) or learning from past HO decisions.

In general, existing HO skipping schemes only consider when the user is moving towards the center of a cell or crossing the edges of one, overlooking other patterns of the user trajectory that could help to further improve the HO decision-making in HWLNets. For example, when one user is walking toward a wall, it is most likely that the LiFi link will be often blocked by its own body. In that case, it is intuitive that such users should be served by WiFi. Otherwise, they will experience too many light-path blockages and HHOs, which shortens their actual available time for transmissions and then degrades their maximum achievable throughput.

3. SYSTEM MODEL

The topology of the HLWNet considered in this work is shown in Fig. 2, where the whole area is divided into square-shaped atto-cells with a LiFi AP in the middle of each of them and the WiFi router is located in a corner of the room. Note that squared LiFi cells provide a performance between the hexagonal and Poisson point process distribution deployments [19], but they involve a simple design, provide good illumination uniformity, and adapt very well to rectangular-shaped rooms. Though the ideal place to install a WiFi router is the center of the coverage area, in practice, the router location is subject to the availability of a network jack and power outlet, and then, in residential environments, it is mostly placed near a wall. We understand that this is the most conventional network topology for realistic HLWNets. Users are deployed randomly around the room, and each of them is holding a user equipment (UE) enabled for both LiFi and WiFi communication. For LiFi, each UE includes V PDs

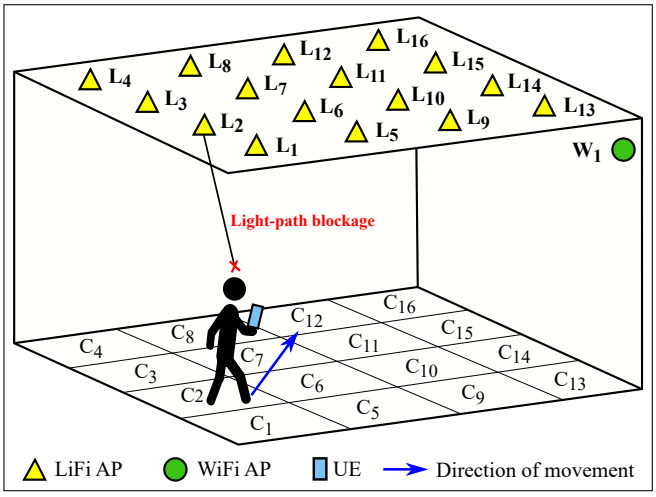

Fig. 2. Considered network topology, where C_i, $i \in \{1, 2, ..., 16\}$, is the atto-cell created by the LiFi AP L_i, and W_1 stands for the WiFi AP.

oriented differently to protect the LiFi link against random UE orientation.

As for the network architecture, note that it consists of an access network where LiFi and WiFi coverage overlap. The WiFi router uses carrier sense multiple access with collision avoidance (CSMA/CA), and the LiFi network employs a reuse factor of 4 to reduce the inter-cell-interference (ICI). Also note that, in practice, neither the UE nor the LiFi or WiFi APs have information about the network topology in a given scenario, i.e., the specific locations of the rest of network elements are typically unknown, as this knowledge would involve the use of sophisticated mapping tools.

A. LiFi Channel Model

The optical path gain from LiFi AP i to the v-th PD of user j is formulated as follows [20]

$$H_{\text{LiFi},v}^{i,j} = \frac{(m+1)A_{\text{PD}}}{2\pi d_{i,j}^2} \cos^m(\phi_{i,j,v}) g_f g_c(\psi_{i,j,v}) \cos(\psi_{i,j,v}) \quad \textbf{(2)}$$

where $m = -ln(2)/ln(\cos(\phi_{1/2}))$ is the Lambertian index of the LED with a half-power semi-angle denoted by $\phi_{1/2}$, A_{PD} is the area of the PD, $d_{i,j}$ is the Euclidean distance between the LiFi AP i and user j, ϕ and ψ are the irradiance and incidence angles, respectively, g_f stands for the gain of the optical filter, and

$$g_c(\psi_{i,j,v}) = \begin{cases} \frac{n^2}{\sin^2 \Psi_{\max}} & 0 \le \psi_{i,j,v} \le \Psi_{\max} \\ 0, & \text{otherwise} \end{cases} \quad \textbf{(3)}$$

is the concentrator gain, where n is the refractive index of the concentrator material and $\Psi_{\max}$ is the maximum incident angle accepted by the concentrator. Then, SINR between LiFi AP i and user j is

$$\gamma_{\text{LiFi}}^{i,j} = \frac{\sum_{v=1}^{V} \left(R_{\text{PD}} H_{\text{LiFi},v}^{i,j} P_{\text{opt}}\right)^2}{N_{\text{LiFi}} B_{\text{LiFi}} + \sum_{\alpha \in \mathcal{I}} \sum_{v=1}^{V} \left(R_{\text{PD}} H_{\text{LiFi},v}^{\alpha,j} P_{\text{opt}}\right)^2} \quad \textbf{(4)}$$

where N_{LiFi} and B_{LiFi} are the power spectral density (PSD) of noise at the receiver and the bandwidth of the LiFi AP, respectively, R_{PD} stands for the PD responsivity, P_{opt} is the optical transmit power per LiFi AP, and $\mathcal{I}$ is the set of interfering APs.

B. WiFi Channel Model

The WiFi channel gain for user j is expressed as [21]

$$G^j_{\text{WiFi}} = |H_{\text{WiFi}}|^2 \, 10^{-\frac{L(d_j)}{10}}, \tag{5}$$

where $L(d_j)$ stands for the path loss of a WiFi channel with the distance between the WiFi router and the user j denoted by d_j and, H_{WiFi}, represents the multipath propagation of a WiFi channel, which is formulated as

$$H_{\text{WiFi}} = \sqrt{\frac{K}{K+1}} e^{j\phi} + \sqrt{\frac{1}{K+1}} X_1, \tag{6}$$

where K is the Ricean K-factor, which is equal to one when d_j is lower than the breaking point distance d_{BP} and zero otherwise, ϕ is the angle of arrival/departure of the line-of-sight component, and X_1 is a complex Gaussian random variable with zero mean and unit variance. The path loss of a WiFi channel is given by [21]

$$L(d_j) = \begin{cases} L_{FS}(d_j) + X_\sigma, & d_j \leq d_{BP} \\ L_{FS}(d_j) + 35 \log_{10}(\frac{d_j}{d_{BP}}) + X_\sigma, & d_j > d_{BP} \end{cases} \tag{7}$$

where X_σ is a zero-mean Gaussian random variable with a standard deviation of σ, and the free-space path loss L_{FS} is defined as follows

$$L_{FS}(d_j) = 20 \log_{10}(d_j) + 20 \log_{f_c}(d_j) - 147.5, \tag{8}$$

where f_c is the operating frequency.

The signal-to-noise ratio (SNR) between the WiFi router and user j is formulated by

$$\gamma^j_{\text{WiFi}} = \frac{G^j_{\text{WiFi}} P_{\text{WiFi}}}{N_{\text{WiFi}} B_{\text{WiFi}}}, \tag{9}$$

where N_{WiFi} and B_{WiFi} are the PSD of noise at the receiver and the bandwidth of the WiFi AP, respectively, and P_{WiFi} is the electrical power transmitted by the WiFi router.

C. Throughput

The HWLNet link capacity for a user j that is being served by either the i-th LiFi AP or the WiFi router can be denoted by

$$r_j = \begin{cases} \frac{B_{\text{LiFi}}}{2} \log_2 \left(1 + \frac{e}{2\pi} \gamma^{i,j}_{\text{LiFi}}\right), & \text{for LiFi} \\ B_{\text{WiFi}} \log_2 \left(1 + \gamma^j_{\text{WiFi}}\right), & \text{for WiFi} \end{cases} \tag{10}$$

Note that, unlike WiFi, LiFi cannot be bounded by Shannon capacity due to modulating the intensity of light, i.e., the transmitted signal is real and positive. Then, we followed up the tighter bound proposed in [22].

D. User Mobility

The user mobility and orientation have been modeled using the orientation-based random waypoint (ORWP) model introduced in [23], where the user moves in zigzag from one waypoint to another, with the waypoints randomly distributed. We assume that the speed is constant along all the way due to the short distances walked indoors [24]. During movement, the orientation of the UE also changes randomly as it occurs in reality [23]. Since the optical wireless channel is not isotropic, the device orientation significantly affects the channel gain and the occurrence of light-path blockages, especially for mobile users.

E. LiFi Line-Of-Sight Link Blockages

Different from the statistical models used in other works in the literature, we assume a realistic object model for the link-path blockage [10]. Specifically, we employ an object model based on cylinders to represent the body of the users within the room. They are modeled as a cylinder of 15 cm radius and 175 cm height and, each of them is holding a UE separated 30 cm from the body at a height of 100 cm.

Considering a more realistic model for the light-path blockages allows us to have a better understanding of the challenges faced by HO algorithms in real HLWNets. For example, we found a high probability of light-path blockage when the user leaves the LiFi cell of its host AP because its own body will likely block the LiFi link, which is illustrated in Fig. 2. This actual effect considerably increases the occurrence rate of light-path blockages in our scenario with respect to other works in the literature that model the blockage as a statistical variable following a Poisson point process [18]. As an example, for a user moving at 3 m/s in our scenario and following the standard handover scheme of LTE [16], we have measured an occurrence rate of 1.5 blockages/s, while the maximum occurrence rate of light-path blockages considered by other works in the literature is 0.07 blockages/s for users moving at 5 m/s [18].

4. NEW HANDOVER ALGORITHM

In this section, we introduce a novel handover scheme that exploits machine learning (ML) techniques (both Supervised Learning and Reinforcement Learning) for HO decision-making in HLWNets. Specifically, we propose an RL-based algorithm that jointly considers (i) the current serving network, (ii) the user's trajectory, and (iii) the user's quality of service in terms of SINR to make optimal HO decisions. To determine the user trajectory, we have designed a prediction model that takes, as inputs, parameters already available for the UE in a realistic scenario (i.e., SINR measurements of all the deployed APs and the ID of the current serving AP).

A. Motivation for using ML techniques and a HO-distributed approach

The main motivation for using RL to make handover decisions in HLWNets is that compared to other approaches in the literature (i.e., trajectory-based [17], optimization-based [25], fuzzy-logic-based [26]), an RL-based HO algorithm (i) does not require previous knowledge of the network topology or exact user trajectory, (ii) reduces the complexity and thus the energy consumption and processing delays, and (iii) enables dynamic adaptation to new network conditions. On the other hand, Supervised Learning has been broadly exploited for prediction/classification tasks, and our motivation to use it in this work is its proven efficiency in predicting user trajectory in mobile networks. Existing works in the literature have shown a prediction accuracy of more than 90% and execution times in the order of millisecond [27].

We adopt a distributed approach to implement the proposed HO algorithm. It means that the HO decisions are made at the UEs and not in a central unit connected to all the deployed WiFi and LiFi APs, as it would be the case for a centralized approach. Our distributed approach presents the following advantages against a centralized one: (i) it requires less signaling between the network elements, which reduces the overall network overhead and minimizes the use of resources; (ii) it reduces the handover delay because the UEs do not need to communicate with the central unit to receive handover-decision instructions;

Algorithm 1. Proposed RL-HO Scheme.

Input: $\gamma_{\mathrm{CH}}^{(t)}, \gamma_{\mathrm{i}}^{(t)}, \forall i \in \mathcal{I}, k \in \{\text{LiFi,WiFi}\}$
Output: i_{NH}
  $t_{\mathrm{c}} \leftarrow 0$
  while $t_{\mathrm{c}} < TTT$ **do**
    if $\gamma_{\mathrm{CH}}^{(t)} + HOM \leq \gamma_{\mathrm{i}}^{(t)}, \forall i \in \mathcal{I}$ **then**
      if k = LiFi & Light-path blockage **then**
        $i_{\mathrm{NH}} \leftarrow$ WiFi AP
        $t_{\mathrm{c}} \leftarrow 0$
      else
        $t_{\mathrm{c}} \leftarrow t - t_0$
    else if $t_{\mathrm{c}} = 0$ **then**
      $t_{\mathrm{c}} \leftarrow t$
    else
      $t_{\mathrm{c}} \leftarrow 0$
  $i_{\mathrm{NH}} \leftarrow$ Output of Algorithm 2.

and (iii) it provides better scalability because its complexity does not increase with the number of nodes in the network.

B. Overview of the Proposed Handover Scheme

Our RL-HO handover scheme is based on RL and assisted by a classification algorithm to predict the user trajectory. It allows (i) finding a near-optimal solution for HO decision-making (i.e., selecting the serving network, either LiFi or WiFi) and dynamically adapting it to new network conditions, (ii) exploiting the user's knowledge about the environment (i.e., current serving network, user trajectory, and user quality of service in terms of SINR) in favor of the HO decision-process, and (iii) learning from the experience of past HO decisions.

Algorithm 1 shows the high-level pseudocode of our proposed RL-HO scheme, where t_0 denotes the time instant when the HO process is triggered. As shown, to trigger the HO decision-making process, we follow a similar approach as in LTE [16], triggering it when any of the available APs $i \in \mathcal{I}$ provides an SINR that is *HOM* dBs larger than the one of the current host AP (γ_{CH}) formulated as:

$$\gamma_{\mathrm{CH}}^{(t)} + HOM \leq \gamma_{\mathrm{i}}^{(t)}, \forall i \in \mathcal{I}. \quad \textbf{(11)}$$

Note that $\mathcal{I}$ is the set of all available APs, i_{CH} is the current host AP, i_{NH} is the new host AP selected by our RL-HO algorithm, and k indicates the current serving network, which could be either LiFi or WiFi.

From Algorithm 1, it should be noticed that, in our RL-HO scheme, when the user is being served by LiFi and the condition in Eq. (11) is satisfied, our algorithm checks if there is a light-path blockage.[2] If so, it immediately switches the user to WiFi and resets the t_{c} counter. By doing so, we avoid outage periods the user would experience if it had to wait for a period *TTT* to make an HO decision, as in the STD-LTE algorithm. Otherwise, if there is no light-path blockage, or the user is being served by WiFi, our algorithm increases the t_{c} counter and waits for a time period *TTT* to make a HO decision. Another important difference with STD-LTE is that when *TTT* is reached, our scheme invokes the RL-based algorithm in Fig. 3 to make the HO decision, instead of just selecting the current target AP as the host one.

[2]Our detection of the light-path blockages is based on measurements of the SINR. We consider there is blockage when the SINR of the host AP LiFi AP is below a certain threshold.

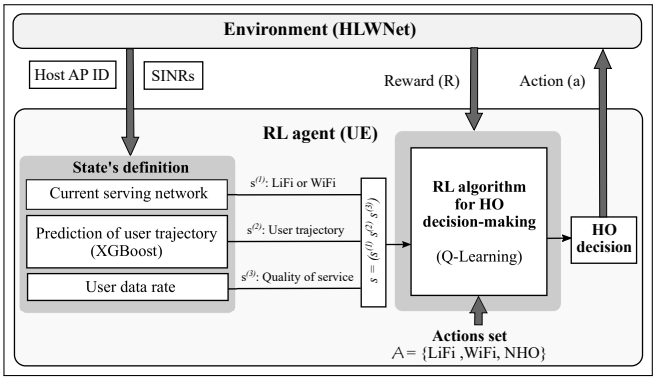

Fig. 3. Block diagram of the proposed RL-HO scheme for HLWNets.

Figure 3 shows the block diagram of the proposed RL algorithm for HO decision-making in HLWNets. As can be observed, it mainly consists of an RL agent that is running at the UE and interacts with the environment (HLWNets) by exchanging certain information. Within the RL agent, we have two main blocks: one to define the states of the RL algorithm and the other to make the HO decisions. As shown, the states of the RL algorithm are defined in terms of the current serving network $k \in \{\text{LiFi, WiFi}\}$, a prediction of the user trajectory, and the quality of service in terms of SINR. For determining the user trajectory, we have designed a prediction algorithm based on Extreme Gradient Boosting trees (XGBoost)[3] whose inputs are (i) the SINR measurements from all the available APs and (ii) the ID of the current host AP. A detailed description of the prediction and RL-based algorithms proposed in this work is provided in Sections 4C and 4D, respectively.

C. Prediction of the user trajectory with Supervised Learning

In this work, we focus on predicting the following cases (classes) of the user trajectory: walking to the center of the cell, crossing a cell edge, or walking toward a wall, which are represented in Fig. 4. These are the most relevant cases for supporting the HO decision-making. In general, if the user is being served by LiFi and crossing the border of a cell, it would be preferable to skip that LiFi AP and switch directly to the next one. This could be learned by the HO algorithm if it had a prediction of the user's trajectory in advance. If the user is connected to the LiFi network and walking toward a wall, most likely it will experience frequent light-path blockages, which will introduce frequent handovers and affect the communication performance. So, for LiFi users walking toward a wall, the optimal choice is to switch them to WiFi to avoid frequent LiFi HHO.

C.1. Proposed prediction model

To predict the user's trajectories, we design the classification algorithm depicted in Fig. 5. To implement it, we opt for an XGBoost-based algorithm [28] because it has already been shown that, among Support Vector Machine (SVM) and a Deep Neural Network (DNN), XGBoost is the one providing the best accuracy on the problem of predicting the user's mobility in cellular networks [27]. Also, note that XGBoost is generally faster and uses fewer computational resources than an ANN. Moreover, XGBoost is designed to be highly parallelizable, making it faster

[3]XGBoost is a variant of the Gradient Boosting Machine (GBM) algorithm where the trees growing is done in a way to reduce the misclassification rate [27].

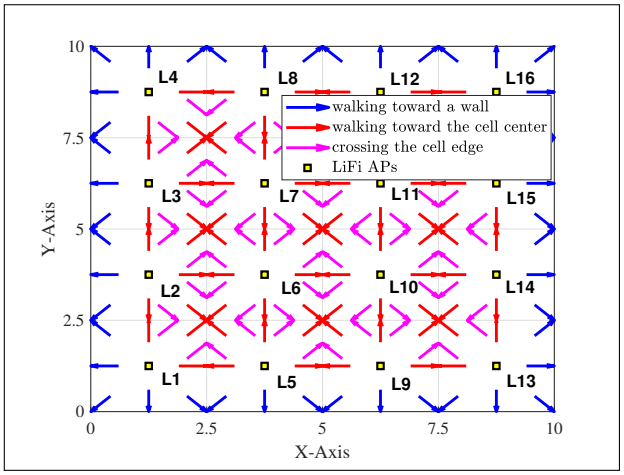

Fig. 4. User's trajectories used for dataset generation.

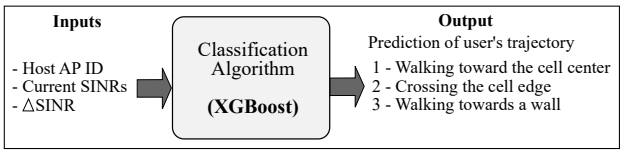

Fig. 5. ML Model for predicting the user's trajectory.

to be trained on large datasets than ANNs, which can be slower to train due to the complexity of the backpropagation algorithm.

As shown, our prediction algorithm takes as input the current SINR measurements for all the APs in the HLWNet and the ID of the current host AP, whereas ΔSINR refers to the SINR difference between the starting ($t_c = t_0$) and ending ($t_c = t_0 + TTT$) points of the t_c counter, so it is derived from the SINR measurements done by the UE. The output of the proposed algorithm will be the prediction of the user's trajectory, for which we use three classes: (1) walking toward the cell center, (2) crossing the cell edge, and (3) walking toward a wall. Note that the selected variables (features) to be used as predictors in the proposed model are highly relevant for predicting the user trajectory. For example, ΔSINR has a high discriminating power because it can indicate whether the user is moving towards the central area of a LiFi AP or not (i.e., a distinctive pattern for class 1). Likewise, users walking towards a wall will measure low SINR values for all the deployed LiFi APs.

C.2. Dataset Generation, Training and Test

We generate the dataset for training the ML model through simulations[4]. Specifically, we simulate a user following the typified trajectories of interest in all the LiFi cells of the considered scenario. For the sake of simplicity, only a subset of the considered trajectories is represented in Fig. 4. For each trajectory within a LiFi cell, we consider different user speeds and starting points relative to the LiFi AP to ensure we generate enough representations of the potential user's location when the HO algorithm is triggered. Specifically, the generated dataset consists of 52800 records considering that the user could be moving at any of the following speeds: 0.5 m/s, 1.0 m/s, 1.5 m/s, 2.0 m/s, 2.5 m/s, or 3.0 m/s, chosen from the average human walking speeds [29] and 200 different start points for each type of user trajectory (i.e., walking towards a wall, walking towards the cell center, or crossing the cell edge) within the LiFi cells.

We use 70% of the dataset for training (following a cross-validation approach with 5 folds) and the remaining 30% for testing. From the hyper-parameter tuning, we got the following optimal parameters for the XGBoost algorithm: learning rate = 0.2, minimum loss reduction = 0.3, maximum depth of a tree = 6, and minimum child weight = 3. The test results of our ML model show an accuracy of 98.5% in predicting the user's trajectory.

[4]Note that, in practice, such dataset can be generated by the assistance of a TurtleBot robot.

D. Handover decision making with Reinforcement Learning

As depicted in Fig. 3, our RL model consists of a set of environment states $\mathcal{S}$, a set of actions $\mathcal{A}$, and a set of rewards $\mathcal{R} \subset \mathbb{R}$, which are defined as follows:

- **States:** The states are defined as the tuple $s = (s^{(1)}, s^{(2)}, s^{(3)}) \in \mathcal{S}$ where $s^{(1)}$ indicates the current serving network, i.e., LiFi or WiFi; $s^{(2)}$ is the prediction of the user trajectory, indicating if it is walking to the center of a cell, crossing the border of one, or walking toward a wall; and $s^{(3)}$ specifies the quality of service in terms of SINR, being a binary variable that takes value 0 or 1 when the SINR of the user is below or above a threshold (set to 10 dB), respectively.

- **Actions**: The actions set is given by $\mathcal{A} = \{\text{LiFi, WiFi, NHO}\}$. Each time the RL algorithm is invoked, the agent selects an action $a \in \mathcal{A}$ which will determine whether the user will be served by the best LiFi AP ($a = $ LiFi), the best WiFi AP ($a = $ WiFi), or do no handover ($a = $ NHO) by skipping it.

- **Reward**: The agent's instantaneous reward is a combination of the handover success, defined as $\mathbb{I} \in \{0, 1\}$ to indicate if the HO was successful or not; the user's satisfaction degree in terms of the maximum achievable data rate, defined as ζ = current data rate/previous data rate ; and the handover cost δ, whose value depends on the type of handover produced by the action taken. Specifically, we set up δ equals 0, 0.3, and 0.7 for no handover (NHO), horizontal handover (HHO), and vertical handover (VHO), respectively, as the VHO is often much more costly than HHO [30, 31]. This way, we encourage the agent to ensure successful handovers and choose the APs that meet its data rate requirements while minimizing the handover overhead. As a result, the reward can be defined as follows:

$$R = \mathbb{I} * (\zeta * \beta + (1 - \delta)(1 - \beta)), \tag{12}$$

where β is a design parameter that controls the tradeoff between guaranteeing the desired data rate and keeping the number of handovers low. If $\beta = 1$, the reward will favor the actions leading to the maximum achievable data rate without considering the number of handovers required to guarantee that. Differently, if $\beta = 0$, the agent will prioritize minimizing the number of handovers at the expense of a lower data rate.

From the above, note that the proposed RL algorithm presents a low complexity. Namely, it has 12 states and 3 actions to choose from in each state. Thus, compared to the alternative of modeling this problem using low-complexity heuristics with fixed rules, our RL-based solution will not offer a much higher complexity, but it ensures the dynamic adaptation to new network conditions.

The aim of our RL algorithm is to find the optimal policy Π^* that maximizes the rewards observed by the user along its trajectory. That is,

$$\Pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbf{E}[R], \tag{13}$$

where $\mathbf{E}$ is the expectation with respect to the randomness in the channel gain.

As depicted in Fig. 3, our RL agent is running on each UE connected to the HLWNet (Environment). Each of them is a decision maker, and their purpose is to find the Π^* that maximizes the cumulative reward over time. Each time the HO algorithm is triggered, the RL agent (i) observes its current state $s \in \mathcal{S}$ by discovering its environment, (ii) makes a HO decision by selecting one of the available actions $a \in \mathcal{A}$, (iii) consequently the UE receives a reward R from the environment and (iv) passes to another state which is defined by the new network conditions. Through this interaction with the environment, the RL agents can learn the optimal HO decisions for each state.

In order to find Π^*, we use the Q-learning because it can provide the simplicity and low computational complexity required to implement the proposed RL-HO algorithm in actual UEs. We adopt Q-learning instead of an On-Policy algorithm (i.e., Sarsa) because it significantly simplifies the modeling, implementation, and analysis of RL-HO. Also, because it guarantees the algorithm convergence as long as all action-state pairs continue to be updated, which we ensure by setting a proper exploration rate. Finally, Q-learning allows learning from the interaction with the environment with no prior knowledge about it (i.e., no need to know the transition probability among the states). Previous works in the literature have already shown the efficiency of Q-learning in solving the HO decision-making problem in heterogeneous networks (HetNets) [32, 33].

In RL, Π^* has an optimal action-value function $q_*(s,a)$ associated with it, which is given by:

$$q_*(s,a) = \max_{\pi} q_\pi(s,a), \text{ for all } s \in \mathcal{S} \text{ and } a \in \mathcal{A}. \tag{14}$$

In Q-learning, q_* is estimated for a specific behavior policy π and for all the considered states s and actions a. Note that, in practice, it is not possible to ensure that the proposed Q-learning algorithm will find q_* and thus make optimal HO decisions because it requires that all state-action pairs are visited in an infinite number of times [34]. Also, the state's definition for the Q-learning algorithm relies on a prediction of the user trajectory that cannot ensure a 100% accuracy. Still, it is possible to show that the Q-learning algorithm actually converges and finds the best action to be taken in a particular state, as we will show in Section 5.

Algorithm 2 depicts the RL algorithm proposed in this work. As can be observed, we follow the ϵ-greedy policy. It means that in step t of the algorithm, the user observes its current state $s_t \in \mathcal{S}$ and takes the action with maximum value (i.e., $a_t = \operatorname{argmax}_{a\in\mathcal{A}} Q(s_t,a)$) with a probability $1-\epsilon$, or a random action with a probability ϵ, where $\epsilon \in [0,1]$ is known as the exploration rate and $Q(s_t,a_t)$ is an estimate of the q_* function in the step t of the RL algorithm. In the next step of the algorithm $(t+1)$, the agent is in a new state ($s_{t+1} \in \mathcal{S}$), where it observes the reward (R_{t+1}) received from the previous action and updates the action-value function using the following equation:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[R_{t+1} + \rho \max_a Q(s_{t+1},a) - Q(s_t,a_t)], \tag{15}$$

Algorithm 2. Proposed Q-learning algorithm for HO decision-making in HLWNets.

Input: Initialize $Q(s,a)\ \forall s \in \mathcal{S}$ and $\forall a \in \mathcal{A}; \epsilon; \alpha; \rho$
Output: a_t

```
t ← 0
while t < t_End do
    Observe s_t
    Generate x from 𝒰[0,1] (uniform distribution)
    if x ≤ ε then
        Uniformly select a random action a_t from 𝒜
    else
        Select the action a_t = argmax_{a∈𝒜} Q(s_t, a)
    t ← t + 1
    Observe s_{t+1} and R_{t+1}
    Update the action-value function using Eq. (15).
```

where α and ρ are two design parameters of the Q-learning algorithm, known as the learning rate and discount rate, respectively. If we look closer at Eq. (15), we notice that the term between brackets is actually the difference between the estimated value $Q(s_t,a_t)$ and its best estimate $R_{t+1} + \rho \max_a Q(s_{t+1},a)$, which can be seen as an error. So, α in Eq. (15) is actually adjusting our estimates based on that error, and ρ indicates how much we value future rewards. Finally, it should be noticed that Algorithm 2 will be executed each time Algorithm 1 invokes it and until the user reaches the end of its trajectory.

5. SIMULATION RESULTS

We use Monte Carlo simulations to evaluate the performance of the proposed method. Our simulation code was implemented using MATLAB and is publicly available to the research community.[5] It consists of two scripts: one to generate the training dataset for trajectory prediction according to Section 4.C.2; and the other to implement our proposed RL-HO algorithm as described in the entire Section 4, in addition to the two baselines considered for comparison in this paper, STE-LTE [16] and Smart HO [18]. Note that both scripts must use the same system model described in Section 3. Finally, to implement the XGBoost algorithm in MATLAB, we used the functions provided in [35].

As shown in Fig. 2, we consider a realistic scenario with only one WiFi AP placed in a corner of the room and 16 LiFi APs separated by 2.5 m. The vertical distance between the user and the LiFi APs is 1.5 m. The bandwidth of all the LiFi APs is set to 20 MHz, which is within the typical range of a 3-dB bandwidth LiFi link in the visible light band [19]. For the WiFi AP, we use 40 MHz, as it is the case of the IEEE 802.11n standard operating in the 5 GHz band. The levels of transmission and noise power, as well as the rest of simulation parameters, are indicated in Table 1. The parameters of the WiFi channel are according to [21], i.e., we set $\phi = 45°$, $d_{BP} = 5$ m, $\sigma = 3$ dB for $d_j \leq d_{BP}$, and $\sigma = 5$ dB for $d_j > d_{BP}$. We consider that each UE is equipped with $V = 2$ PDs placed in orthogonal planes [36]. The average overhead of HHO and VHO are set to 200 ms and 500 ms, respectively [31]. To provide a fair comparison, all the compared algorithms use exactly the same parameters' values, including the same *HOM* and *TTT*, which are set to 1 dB and 320 ms, respectively [16]. Also, all of them immediately switch the user to WiFi in case of light-path blockage. Unless we indicate otherwise, for the RL algorithm we set ϵ, α, and ρ to 0.1, 0.5, and 0.2, respectively.

[5] https://github.com/dayrenefrometa/RL-HO

Parameter	Value
Room size (length by width by height)	10 x 10 x 3 m
The physical area of each PD, A_{pd}	1 cm^2
Optical filter gain, g_f	1
Refractive index, n	1
FoV semi-angle of the PD, Ψ_{max}	90°
Detector responsivity, R_{pd}	0.53 A/W
Half-intensity radiation angle, $\Phi_{1/2}$	60°
Modulated optical power per LiFi AP, P_{opt}	3.5 W
Transmitted power of the WiFi AP, P_{WiFi}	15 dBm
Bandwidth per LiFi AP, B_{LiFi}	20 MHz
Bandwidth per WiFi AP, B_{WiFi}	40 MHz
PSD of noise in LiFi, N_{LiFi}	10^{-21} A^2/Hz [37]
PSD of noise in WiFi, N_{WiFi}	-174 dBm/Hz [37]

Table 1. Simulation parameters.

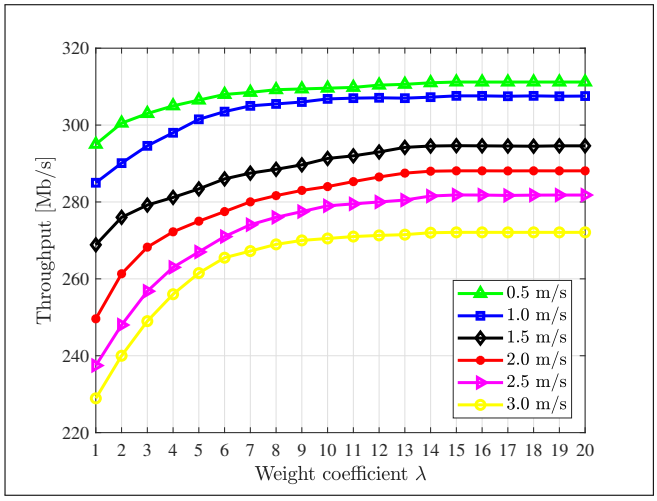

Fig. 6. User throughput versus the weight coefficient λ of Smart HO.

A. Finding the optimal λ of Smart HO in our scenario

One of the baselines used in this paper is the Smart HO algorithm [18] described in Section 2B, which uses the objective function in Eq. (1) to determine the target AP. One key parameter of the objective function is the weight coefficient λ that should be optimized according to the user's speed. Also, we should notice that the optimal values of λ highly depend on the considered scenario. In this work, we analyze a different scenario from the one in [18]. So, in order to not underestimate the results and make a fair comparison, we first find the optimal values of λ to be used in the baseline Smart HO algorithm under the new scenario.

Fig. 6 shows the user's throughput as a function of the weight coefficient λ. As it can be observed, for our scenario, the throughput saturates at a certain λ value, and it is possible to find λ providing a near-to-optimal performance for all the considered user speeds. Specifically, for $\lambda = 15$, we can get the maximum achievable throughput for each user speed. For that reason, in all the simulation results presented in this section, we set λ=15 regardless of the user speed. Also, this guarantees a more practical approach and ensures a fair comparison with our RL-HO algorithm, which does not use information about the user's speed to make the handover decisions.

B. Handover rate

Fig. 7 shows the rates of HHO and VHO for different user speeds. In general, given the small coverage of the LiFi cells, the handover rate for all the considered approaches increases with the user speed. However, we can observe that the proposed method ensures the lowest rates of HHO and VHO for most of the considered user speeds, while STD-LTE provides the highest ones. The only exception is for a user speed of 0.5. m/s when our approach provides a slightly larger rate of HHO than the STD-LTE or Smart HO approaches. This is because, for slow-moving users, our RL-HO understands it is better to keep the user connected to the LiFi network, which could lead to frequent HHO. Nonetheless, as depicted in Fig. 7, the HHO rate increase with respect to the other approaches is marginal. Also, it should be noted that, for our scheme, the increase of the handover rate with the user speed is slower than for the other approaches. This is because by exploiting the prediction of the user trajectory, our RL-HO algorithm can learn to skip unnecessary handovers, especially those triggered when the user is crossing the edge of a cell. More specifically, for a user speed of 3 m/s, RL-HO can reduce the rate of VHO of STD-LTE and Smart HO at 54% and 43%, respectively. Likewise, the rate of HHO is reduced at 85% and 71%, with respect to STD-LTE and Smart HO, respectively. Fig. 8 depicts the percentage of time that the user is connected to each network (LiFi or WiFi) or under light-path blockage, for each handover approach and different user speeds. As shown, when the user is moving fast, our RL-HO algorithm decides to serve it with WiFi most of the time (more than 80% of the time for 3 m/s), which considerably reduces the handover rate as mentioned before.

Finally, note that in our simulation results, the rate of VHO is much higher than the HHO one for all the considered user speeds and handover mechanisms. This is because we are working with a realistic LiFi link blockage model that leads to a high occurrence rate of blockages, as shown in Figure 9. Since the user is in movement all the time, and given the hysteresis principle followed by all the considered HO approaches, each time the user is transferred to WiFi due to a light-path blockage (which happens really often), it is hard for the HO algorithms to switch the user back to LiFi and keep it connected to the LiFi network for a long time, especially when it is moving fast.

C. Average throughput

Fig. 10 presents the user throughput as a function of the user's speed. As shown, in this case, we have tested the proposed RL-HO algorithm with different exploration rates (i.e., $\epsilon = 0.01$, $\epsilon = 0.1$, $\epsilon = 0.3$, and $\epsilon = 0.7$). In all cases, our algorithm significantly outperforms STD-LTE and Smart HO for all the considered user speeds. Specifically, for a user speed of 3 m/s, our RL-HO with $\epsilon = 0.1$ can provide an average throughput that is 1.6 and 2.5 times higher than the ones provided by Smart HO and STD-LTE, respectively. Also, note that, unlike the other approaches, for $\epsilon = 0.1$, $\epsilon = 0.3$, and $\epsilon = 0.7$, the proposed RL-HO algorithm provides an average throughput that increases with the user speed. This is because our RL-HO algorithm learns by (1) being triggered, (2) making an HO decision, and (3) receiving the corresponding reward. So, the higher the user speed, the more likely the RL algorithm is triggered due to variations on the WiFi and LiFi channels, and thus, the faster it

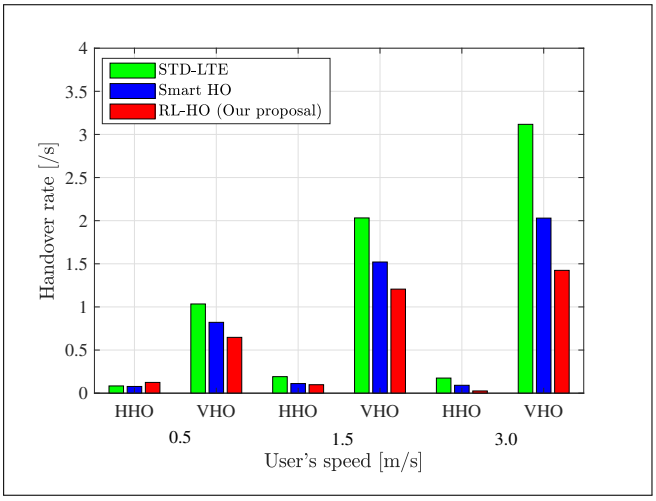

Fig. 7. Rates of HHO and VHO.

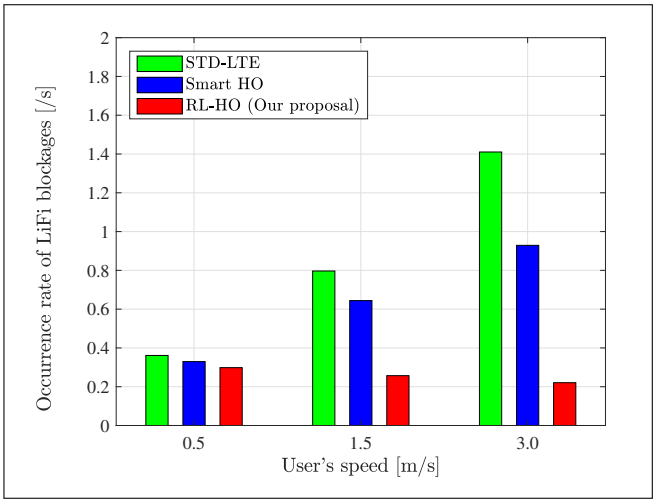

Fig. 9. Occurrence rate of LiFi blockages.

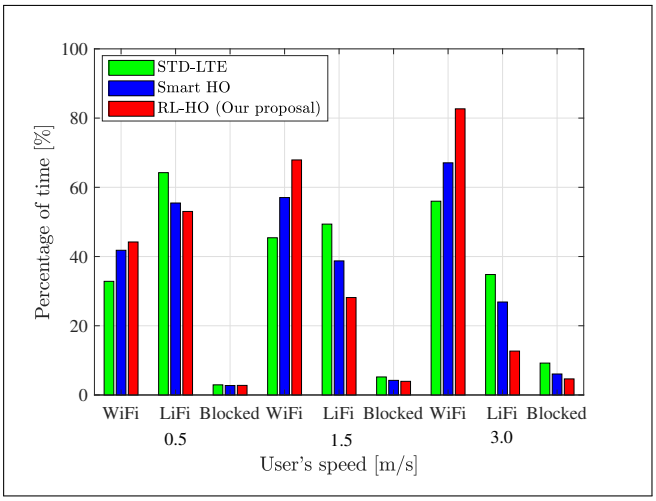

Fig. 8. Percentage of time the user is connected to WiFi, LiFi, or blocked along its trajectory.

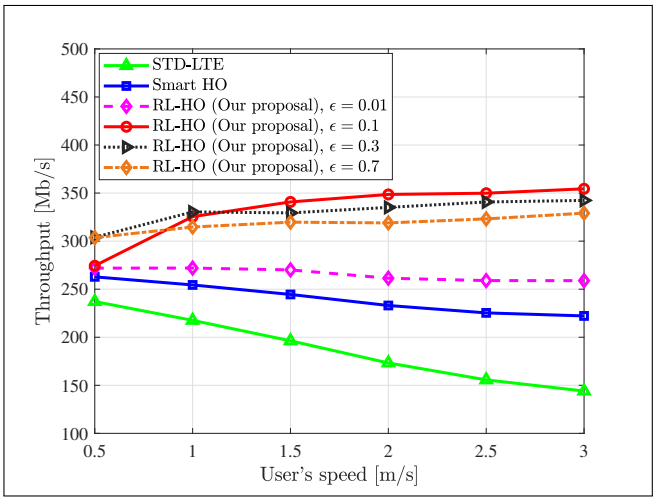

Fig. 10. Average throughput versus the user's speed.

learns that it is better to serve fast-moving users with WiFi. Each time the RL algorithm decides to switch a fast-moving user to the LiFi network, it will likely find a light-path blockage or a non-satisfied user, thus receiving a low reward. Quickly learning that it is better to serve fast-moving users with WiFi will oftentimes reduce the handover rate and light-path blockage probability, which considerably improves the average throughput, as shown in Fig. 10.

From Fig. 10, we should also note that by slightly increasing the exploration rate ($\epsilon = 0.3$), we can improve the average throughput for slow-moving users, while for fast-moving users, it is better to keep the exploration rate low ($\epsilon = 0.1$). Also, notice that $\epsilon = 0.01$ worsens the performance of our RL-HO mechanism. The reason is that with such a lower exploration rate, our RL algorithm keeps exploiting the same actions instead of exploring others that could provide a better performance. Finally, note that for $\epsilon = 0.7$, the proposed RL algorithm shows lower performance than for $\epsilon = 0.3$, which is the consequence of increasing the probability of selecting random actions. We did not try exploration rates larger than 0.7 because it means selecting random actions very often, which will worsen the performance even more.

D. Impact of the Number of Users

Finally, we evaluate the impact of the number of users considered within the room. Fig. 11 shows the average throughput of one user along its trajectory when increasing the total number of users walking around the same room. As shown, the average throughput of STD-LTE and Smart HO decreases with the number of users because more users walking around will increase the probability of light-path blockage. However, the proposed RL-HO algorithm, provides an almost constant average throughput. We can even see a slight increase with the number of users. This is because the proposed RL algorithm learns by interacting with the environment and updating the action-value function each time it is triggered. So, increasing the number of users increases the number of light-path blockages and, thus, the number of times the RL algorithm is triggered, boosting up the learning process to find the best action for a particular state. In other words, increasing the number of users will quickly teach our RL algorithm the best handover decisions it could make, slightly increasing the average user throughput.

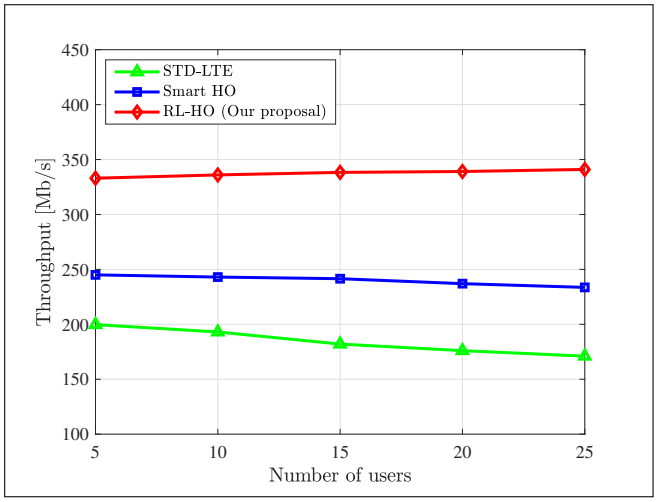

Fig. 11. Average achievable throughput versus the number of users for a user speed of 1.5 m/s.

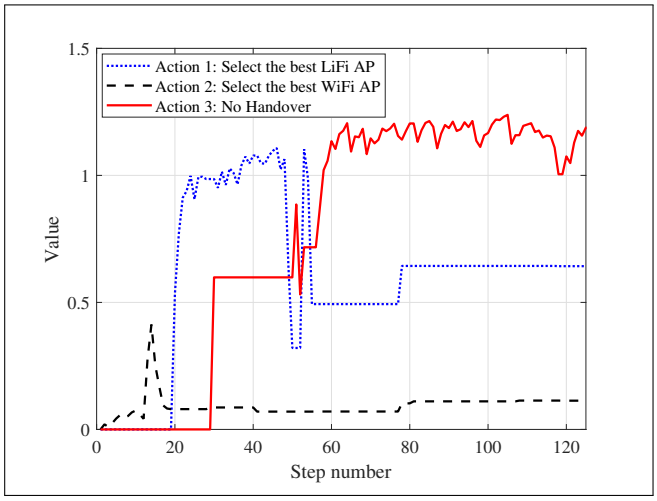

Fig. 12. Action values vs. step number for state S, where the step number indicates the number of times that the RL algorithm has been triggered under state S.

E. Convergence of the proposed RL algorithm

Figure 12 provides more details about the learning process of the proposed RL algorithm and its convergence time (i.e., how long it takes to find the best action to be taken in a specific state). For a certain state S (user served by LiFi, crossing the edge of a LiFi cell, and receiving good SINR), Figure 12 shows the action values for each step number of the RL algorithm. Note that by step number, we mean the number of times the RL algorithm has been triggered under state S. As observed, after around 60 steps, the RL algorithm converges because it has found the best action for this particular state. Action A3 (No Handover) shows and maintains the highest value compared to the other actions, which makes sense because, in this particular state, it sounds reasonable to avoid increasing the network overhead with unnecessary handovers. Finally, note that although the convergence time may seem long, once the RL agent is trained, the optimal configuration of the RL algorithm can be shared with new UEs joining the network.

6. CONCLUSION

In this work, a novel ML-based handover scheme for HLWNets was proposed. It uses the SINR measurements already available at the user equipment to predict the user trajectory and supports a Reinforcement Learning algorithm that makes the handover decisions. Unlike other approaches that use the rate of change of the SINR to determine if the user is moving toward the center of a cell or not, we implement a classification algorithm that allows us to predict more patterns of the user trajectory (e.g., user walking to a wall), which we exploit to optimize the handover decision-making process. Our RL-HO scheme consists of a reinforcement learning agent that makes the handover decisions based on the current state, and by interacting with the HLWNet, it learns how to make near-optimal handover decisions. Specifically, it learns that fast-moving users or those walking to a wall are better served by WiFi in order to avoid frequent unnecessary handovers. On the other hand, it learns that slow-moving users are better served by LiFi and that the LiFi APs that the users cross by their cell edge can be skipped to minimize unnecessary handovers. Simulation results have shown that RL-HO reduces the handover rate of the standard HO scheme of LTE by 54% at a user speed of 3 m/s. Regarding the average throughput along the user's trajectory, RL-HO improves the ones provided by the standard handover scheme of LTE and the smart handover algorithm by 146% and 59%, respectively.

ACKNOWLEDGMENTS

This work has been partially funded by European Union's Horizon 2020 Marie Sklodowska Curie grant ENLIGHT'EM (814215), in part by the MSCA Postdoctoral Fellowship grant RISA-VLC (101061853), and in part by the project RISC-6G, reference TSI-063000-2021-59, granted by the Ministry of Economic Affairs and Digital Transformation and the European Union-NextGenerationEU through the UNICO-5G R&D Program of the Spanish Recovery, Transformation and Resilience Plan.

DISCLOSURES

The authors declare no conflicts of interest.

REFERENCES

1. C. Mas-Machuca, M. Kaufmann, J.-P. Linnartz, M. Riegel, D. Schulz, and V. Jungnickel, "Techno-economic study of very dense optical wireless access using visible or infrared light," J. Opt. Commun. Netw. **15**, B33–B41 (2023).
2. B. Makki, T. Svensson, K. Buisman, J. Perez, and M.-S. Alouini, "Wireless Energy and Information Transmission in FSO and RF-FSO Links," IEEE Wirel. Commun. Lett. **7**, 90–93 (2018).
3. M. Halper, "Li-Fi's latest champion: Cisco," LEDs Mag. (2022).
4. H. Haas, L. Yin, Y. Wang, and C. Chen, "What is LiFi?" J. Light. Technol. **34**, 1533–1544 (2016).
5. S. Pergoloni, M. Biagi, S. Colonnese, R. Cusani, and G. Scarano, "Optimized LEDs footprinting for indoor visible light communication networks," IEEE Photonics Technol. Lett. **28**, 532–535 (2016).
6. M. D. Soltani, H. Kazemi, M. Safari, and H. Haas, "Handover modeling for indoor Li-Fi cellular networks: The effects of receiver mobility and rotation," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, (2017), pp. 1–6.
7. X. Wu and H. Haas, "Handover skipping for LiFi," IEEE Access **7**, 38369–38378 (2019).
8. E. Demarchou, C. Psomas, and I. Krikidis, "Mobility Management in Ultra-Dense Networks: Handover Skipping Techniques," IEEE Access **6**, 11921–11930 (2018).

9. A. Ford, C. Raiciu, M. J. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," RFC 6824 (2013).
10. X. Wu, M. D. Soltani, L. Zhou, M. Safari, and H. Haas, "Hybrid LiFi and WiFi networks: A survey," IEEE Commun. Surv. & Tutorials **23**, 1398–1420 (2021).
11. Y. Wang, X. Wu, and H. Haas, "Fuzzy logic based dynamic handover scheme for indoor Li-Fi and RF hybrid network," in *2016 IEEE International Conference on Communications (ICC),* (2016), pp. 1–6.
12. X. Wu and D. C. O'Brien, "A novel machine learning-based handover scheme for hybrid LiFi and WiFi networks," in *2020 IEEE Globecom Workshops (GC Wkshps,* (2020), pp. 1–5.
13. F. Wang, Z. Wang, C. Qian, L. Dai, and Z. Yang, "Efficient vertical handover scheme for heterogeneous VLC-RF systems," J. Opt. Commun. Netw. **7**, 1172–1180 (2015).
14. S. Liang, H. Tian, B. Fan, and R. Bai, "A novel vertical handover algorithm in a hybrid visible light communication and LTE system," in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall),* (2015), pp. 1–5.
15. Y. Wang, X. Wu, and H. Haas, "Load Balancing Game With Shadowing Effect for Indoor Hybrid LiFi/RF Networks," IEEE Transactions on Wirel. Commun. **16**, 2366–2378 (2017).
16. "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RCC); Protocol Specification (Release 13)," Standard 3GPP TS 36.331 v13.0.0., Tech. Rep., LTE, Valbonne, France (2016).
17. R. Arshad, H. Elsawy, S. Sorour, T. Y. Al-Naffouri, and M.-S. Alouini, "Handover management in 5G and beyond: A topology aware skipping approach," IEEE Access **4**, 9073–9081 (2016).
18. X. Wu, D. C. O'Brien, X. Deng, and J.-P. M. G. Linnartz, "Smart handover for hybrid LiFi and WiFi networks," IEEE Transactions on Wirel. Commun. **19**, 8211–8219 (2020).
19. C. Chen, D. A. Basnayaka, and H. Haas, "Downlink performance of optical attocell networks," J. Light. Technol. **34**, 137–156 (2016).
20. J. Kahn and J. Barry, "Wireless infrared communications," Proc. IEEE **85**, 265–298 (1997).
21. E. Perahia and R. Stacey, *Next Generation Wireless LANs: 802.11n and 802.11ac* (Cambridge University Press, 2013), 2nd ed.
22. J.-B. Wang, Q.-S. Hu, J. Wang, M. Chen, and J.-Y. Wang, "Tight bounds on channel capacity for dimmable visible light communications," J. Light. Technol. **31**, 3771–3779 (2013).
23. M. D. Soltani, A. A. Purwita, Z. Zeng, H. Haas, and M. Safari, "Modeling the random orientation of mobile devices: measurement, analysis and LiFi use case," IEEE Transactions on Commun. **67**, 2157–2172 (2019).
24. X. Wu and H. Haas, "Mobility-aware load balancing for hybrid LiFi and WiFi networks," Journal of Optical Communications and Networking. **11**, 588–597 (2019).
25. X. Li, R. Zhang, and L. Hanzo, "Cooperative Load Balancing in Hybrid Visible Light Communications and WiFi," IEEE Transactions on Commun. **63**, 1319–1329 (2015).
26. X. Wu, M. Safari, and H. Haas, "Access Point Selection for Hybrid Li-Fi and Wi-Fi Networks," IEEE Transactions on Commun. **65**, 5375–5385 (2017).
27. H. Gebrie, H. Farooq, and A. Imran, "What machine learning predictor performs best for mobility prediction in cellular networks?" in *2019 IEEE International Conference on Communications Workshops (ICC Workshops),* (2019), pp. 1–6.
28. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* (Association for Computing Machinery, New York, NY, USA, 2016), KDD '16, p. 785–794.
29. R. W. Bohannon, "Comfortable and maximum walking speed of adults aged 20-79 years: reference values and determinants," Age Ageing **26**, 15–19 (1997).
30. "A Pre-Scanning Fast Handoff Scheme for VoIP in WLANs, author=Jun Xiao and Feng Liu," (2015). Available at: https://api.semanticscholar.org/CorpusID:61291048.
31. H. Kwon, K.-Y. Cheon, and A. Park, "Analysis of WLAN to UMTS handover," in *2007 IEEE 66th Vehicular Technology Conference,* (2007), pp. 184–188.
32. E. Fakhfakh and S. Hamouda, "Incentive reward for efficient WiFi offloading using Q-learning approach," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC),* (2017), pp. 1114–1119.
33. S. Khosravi, H. Shokri-Ghadikolaei, and M. Petrova, "Learning-based handover in mobile millimeter-wave networks," IEEE Transactions on Cogn. Commun. Netw. **7**, 663–674 (2021).
34. R. S. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, Adaptive computation and machine learning (The MIT Press, Cambridge, Massachusetts, 2014), nachdruck ed.
35. J. van Prehn, "Functions to run xgboost in Matlab," Available at: https://es.mathworks.com/matlabcentral/fileexchange/75898-functions-to-run-xgboost-in-matlab.
36. M. D. Soltani, M. A. Arfaoui, I. Tavakkolnia, A. Ghrayeb, M. Safari, C. M. Assi, M. O. Hasna, and H. Haas, "Bidirectional optical spatial modulation for mobile users: Toward a practical design for LiFi systems," IEEE J. on Sel. Areas Commun. **37**, 2069–2086 (2019).
37. X. Wu and D. O'Brien, "A novel handover scheme for hybrid LiFi and WiFi networks," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC),* (2020), pp. 1–5.