Resource Allocation Using Deep Learning in Mobile Small Cell Networks

Saniya Zafar, Sobia Jangsher, Member, IEEE, and Arafat Al-Dweik, Senior Member, IEEE

Abstract—This work proposes a position-dependent deep learning (DL)-based algorithm that enables interference free resource allocation (RA) among mobile small cells (mScs). The proposed algorithm considers a vehicular environment comprising of city buses that generates historic data about the city buses positions. The position information of the moving buses is exploited to form interference free resource block (RB) allocation as data labels to the respective historic data. The long short-term memory (LSTM) algorithm is used for RA in mSc network based on position-dependent historic data. The numerical results obtained under non-dense and dense mSc network scenarios reveal that the proposed algorithm outperforms other machine learning (ML) and DL-based RA mechanisms. Moreover, the proposed RA algorithm shows improved results when compared to RA using Global Positioning System Dependent Interference Graph (GPS-DIG), but provides less data rates as compared to existing Time Interval Dependent Interference Graph (TIDIG)-based, and Threshold Percentage Dependent Interference Graph (TPDIG)-based RA while fulfilling the users’ demands. The proposed scheme is computationally less expensive in comparison with TIDIG and TPDIG-based algorithms.

Index Terms—Deep learning (DL), long short-term memory (LSTM), mobile-small cells (mScs), resource allocation (RA), resource blocks (RBs), fifth generation (5G), sixth generation (6G).

I. INTRODUCTION

MOBILE communication systems were initially introduced to provide voice-only services to a limited number of user. For example, first generation (1G) networks have supported about 11 million users worldwide in 1990 [1]. Currently, wireless networks such as fourth generation (4G) and fifth generation (5G) support more than 6 billion users [2], [3], i.e., about 550 fold increase. Nevertheless, the increased number of users is manageable for voice-only services because of the limited bandwidth, power, and quality-of-service (QoS) requirements for voice centric applications. The main challenge, in fact, is the enormous growth of data traffic caused by the increased number of subscribers, ultra-high data rates, and deployment of new applications such as Internet of Things (IoT) and machine-to-machine (M2M) communications. Some recent statistics show that mobile data traffic has increased by 68% in 2019 reaching 38 Exabytes (EB)/month as compared to 27 EB/month in 2018 [2], [3]. Moreover, Ericsson anticipates that mobile data traffic will annually increase by 27% in 2025 [2]. Consequently, future wireless networks should be optimized at all levels to be able to satisfy the escalating demands for data and high QoS requirements. Network densification and efficient resource allocation (RA) are among the prominent techniques to be adopted for the sixth generation (6G) wireless networks to increase the network efficiency [4].

Network densification is considered as a pivotal solution for increasing the network capacity in wireless systems, and it is considered essential in 5G network architecture [5]. The primary concept of network densification is to deploy a large number of low power small-cells to complement the macro-cell functionality in regions suffering from low signal quality such as indoor and highly dense areas [6], [7]. Small cells have different coverage capabilities and are typically classified as femto, mini, pico and micro cells. When a small-cell is incorporated in a vehicle, it is denoted as mobile-Small cell (mSc). One of the main advantages of deploying mScs is that they mitigate the significant attenuation of wireless signals when they propagate through vehicle body, which is typically made of thick metal. Moreover, the distance between mSc access point (mScAP) and its associated mobile users is decreased, which in turn enhances the QoS inside the vehicle [7]–[9].

RA aims at distributing the transmission resources among different network users, and if the RA strategy is performed optimally, it may significantly improve the network performance in terms of capacity and user satisfaction [6], [10]. The synergy of network densification and RA would be highly beneficial, particularly when the RA is optimized specifically for small-cell networks. Therefore, RA for small-cell networks have received significant interest by the research community as reported in [4], [11], [12], and the references listed therein. Moreover, RA for mSc networks has been considered as described in [7], [13], [14].

Unlike macro-cell and static small-cell networks, RA for mScs is more challenging because of the time-varying channel conditions [15] and the high handover frequency between...
mScs and macro-cells as well as the mobile users and mScs [13]. The cell mobility in dense cellular networks also introduces irregular time-varying interference patterns between mScs in close proximity of each other. Consequently, satisfying the demands of users connected to mScs typically requires allocating more transmission resources as compared to static small-cell networks, and the RA has to be performed more frequently. Therefore, adopting efficient RA strategies is indispensable under such stringent constraints.

A. Related Work

Optimal RA in an hierarchical network with static and mobile small cell access points is a challenging problem. The problem with static small cells have been thoroughly investigated using game theory and various optimization techniques in centralized [6], [10], [12], [16] and distributed manner [17], [18]. Along with static small cells, some work on RA for mobile small cells is also present in the literature based on mathematical programming which involves either maximizing the sum-rate or minimizing the interference between mScs in a network [7], [9], [19]. For example, Xiao et al. [17] proposed a non-cooperative game theory mechanism that considers the vehicular mobility characteristics for RA in vehicular heterogeneous networks. The work in [7]–[9] considers RA in vehicles travelling at a uniform speed along a fixed path, and [19] consider the case of variable speed vehicles following a predefined path.

Despite of the compelling performance of traditional centralized and distributed approaches, there are certain challenges that have to be considered. For example, the centralized mechanisms induce increased signaling overhead with significant computational complexity whereas distributed mechanisms have reduced implementation complexity as compared to centralized algorithms. However, the performance of the distributed is inferior as compared to centralized as well as de-centralized approaches [6]. Although the scalability to large-scale network of the distributed algorithm is better as compared to the centralized approaches, it is still challenging to implement it for highly dense network. In comparison with aforementioned traditional mechanisms, real-time implementation of DL algorithms both centralized and distributed can be done with reduced complexity [20]. DL-based techniques can be applied to directly learn the required information using the historic data instead of less compliant existing mathematical models.

DL is a powerful data driven mechanism, it can be applied for RA to solve complex optimization problems in real-time [20]. Wu et al. [21] described a mechanism for empowering mobile devices that are capable of adapting dynamic RA using commercial-off-the-shelf based DL systems. In [22], authors addressed the RA problem in wireless networks using deep reinforcement learning. The authors proposed circumstance (number of users and QoS requirements) independent mechanism that can efficiently deal with various circumstances of network. Zhao et al. [23] described a distributed deep reinforcement learning-based platform for optimal association of user equipment to base stations and allocation of radio resources to user terminals in heterogeneous networks. The authors of [24] discussed the resource sharing problem from virtual mobile (VM) operators’ perspective in software defined networking dependent VM network empowered by renewable energy. The work in [25] focuses on development of vehicular edge computing network where vehicles serve as edge computing servers for user devices. Reinforcement learning is utilized by the authors to obtain optimal RA to vehicular edge computing systems.

Double deep Q-learning network-based allocation of computation as well as communication resources is proposed in [26] for edge computing networks comprising of a number of edge servers and mobile devices to achieve low latency and balancing the load. The authors of [27] presented a DL-based hybridized model for space-time prediction by proposing deep learning-based auto encoder to model spatial prediction and Long Short-Term Memory (LSTM) network for temporal modeling. Furthermore, [28], [29] considered LSTM for future traffic prediction in radio access network slicing to deal with the resources allocation. Moreover, LSTM-based prediction of RA to individual slice in sliced radio access network is presented in [30] to reduce the system delay.

Similarly, from the perspective of vehicular networks, vehicles associated to IoTs with high performance storage and computation capabilities along with numerous on-board sensors (including cameras, radars etc.), are participating in generation and collection of data followed by storage, processing, and transmission of vast amounts of data to make the driving experience convenient and safer [31]. The mentioned sources of data drive us to explore the new dimensions in designing an efficient and reliable moving networks. The research work in vehicular networks have demonstrated that significant performance gain can be achieved by adopting DL. For example, Liang et al. [20] provides a comprehensive review unleashing the potentials of DL and deep reinforcement learning-based RA over traditional RA mechanisms used in wireless communications. The authors also highlighted the potential of DL for RA problems in generalized vehicle-to-vehicle (V2V) communications networks. In [32], autonomous allocation of resource blocks (RBs) and transmission power using multi agent deep reinforcement learning is presented for train-to-train communications, and communications between vehicles travelling with deterministic speeds. The authors of [13], [14] examined the allocation of resources in a mSc network and mobile Internet-of-Things (mIoT-mSC) networks, respectively. More specifically, [13] targeted optimizing the RB allocation by minimizing the interference between mScs, while [14] considered maximizing the data rates in mIoT-mSC networks. In these works, DL is used as an initial step for interference determination between the mScs, and later, the resources are allocated using mathematical approaches that require accurate and real-time mSc information. The above analysis motivate us to explore RA using DL in probabilistic mobility-based scenarios as it is not yet investigated in the literature [33].
B. Motivation and Main Contributions

With the increasing demand for data connectivity over various transportation systems, it is becoming apparent that using conventional RA methods will not be able to cope with stringent time and complexity constraints inherent to mobile systems. In particular, the use of mathematical programming in conventional RA methods will incur computational time that can be prohibitive for mobile applications [10], [34], [35]. Therefore, DL can be used to overcome the shortcomings of conventional RA mechanisms.

Based on the surveyed literature, the references listed therein, and to the best of the authors’ knowledge, there is no work in the open literature that considers DL for mobile small cell networks. Although the work in [14] considers DL for interference management as applied to RA, DL is used only to determine the mScs locations, which are then used to form interference patterns that can be used for RA. Therefore, we propose in this work an efficient position-dependent (PD) LSTM algorithm, called PD-LSTM, for RB allocation in a mSc network formed for a city bus transportation system. Overall, the contributions of the paper can be briefly summarized as follows.

- Formulate a RA problem in mSc network to allocate resources to mScs moving with non-uniform speed.
- Examine the deployment of mScs in nearly realistic scenario of city buses associated to different bus routes.
- Formulate a data set comprising of spectrum data [31] using the vehicular network tool box and utilize graph theory [36] to optimally allocate the RBs to all mScs based on their positions at each time instant to achieve ground truth labels.
- Propose a position-dependent LSTM [37] algorithm for RA in mSc network such that the demand of each cellular user is fulfilled.
- Numerical results of proposed RA method are compared with DL-based and conventional RA mechanisms under non-dense mSc network and dense mSc network scenarios.

C. Paper Structure

The rest of the paper is arranged as: Section II explains the system model of mSc network and Section III specifies the optimization problem. Section IV proposes the DL-based resource allocation method and Section V demonstrates the respective simulation results. Section VI provides the conclusion of paper.

II. SYSTEM MODEL

This work considers a dense cellular network comprises of a macro-cell served by a macro base station (BS) and a number of mScs served by small wireless access points, as depicted in Fig. 1. The mScs are considered to be deployed in city buses that move along pre-defined paths with variable speed. The network model considers the incorporation of $M$ mScs in $B$ city buses travelling along $C$ routes, $B = \{1, 2, \ldots, B\}$ represents the set of city buses. The total number of buses moving on route $r$ is denoted as $B_r$ where $r \in \{1, 2, \ldots, C\}$. The letter $j$ is used to denote mSc in a network, which are then used to form interference patterns that can be used for RA. Therefore, we propose in this work an efficient position-dependent (PD) LSTM algorithm, called PD-LSTM, for RB allocation in a mSc network formed for a city bus transportation system. Overall, the contributions of the paper can be briefly summarized as follows.

- Formulate a RA problem in mSc network to allocate resources to mScs moving with non-uniform speed.
- Examine the deployment of mScs in nearly realistic scenario of city buses associated to different bus routes.
- Formulate a data set comprising of spectrum data [31] using the vehicular network tool box and utilize graph theory [36] to optimally allocate the RBs to all mScs based on their positions at each time instant to achieve ground truth labels.
- Propose a position-dependent LSTM [37] algorithm for RA in mSc network such that the demand of each cellular user is fulfilled.
TABLE I
SYSTEM MODEL PARAMETERS

| Parameter | Description |
|-----------|-------------|
| M         | Total mobile Small cells in a network |
| B         | Set of bus routes |
| C         | Set of city bus routes |
| U         | Total mobile users in a network |
| J         | Total resource blocks |
| D         | Minimum demand of a mSc |
| T         | Transmission time frame |
| N         | Total time slots in period |
| t         | Time slot |
| Δt        | Length of a time slot |
| Z         | Interference set associated to a mSc j |
| Z^k_{u,j} | RA indicator to manage RB allocation to user u of mSc j |
| h^k_{u,j} | Channel gain of user u associated to a mSc j over RB k |
| d^k_{u,j} | Distance between user u and mSc j |

Fig. 2. Time slot formation in time period.

different mSc access links. Therefore, if one or more mScs are using the same RB, such mScs will have a mutual interference if they are within the coverage range of each other.

The mScs in a network are considered using orthogonal frequency division multiple access (OFDMA) [38], where the transmission time and frequency are divided into time-frequency units as described in [10]. Each time-frequency unit is considered as a RB, which is the smallest resource unit that can be allocated to any mSc. The network is assumed to have total of K RBs that should be assigned to M mScs. The set of RBs is defined as K = {1, 2, ..., K}. Moreover, all RBs will be allocated similar transmit power, and the minimum number of RBs of each mSc is denoted as D_j. Table I provides the system parameters and their description. The transmission time frame T is divided into N equal time slots t_1, t_2, ..., t_N, where each slots has length Δt. Fig. 2 shows the time slots formation in a given time period, and the channel is considered as quasi-static in each time slot.

A transmission is considered successful for the access link of the mSc if the signal to interference-plus-noise ratio is above a defined threshold. A protocol interference model is considered [39], in which the two transmissions will interfere if they are present in each others’ interference range. Therefore, the interference between mScs mainly depends on the position of the buses, which can be determined using their speed and route information. Consequently, all mScs that exist within the interference range of mSc j are the interfering mScs, and form an interference set I_j for mSc j. The signal to interference-plus-noise ratio of user u connected to mSc j and RB k in the presence of interference set I_j in time slot t_i is given by

\[
gamma^k_{u,j}(t_i) = \frac{\sum_{m=1}^M z^k_{u,m}(t_i) p^k_{u,m}(t_i) d^{-β}_{u,m}(t_i)}{N_0 + \sum_{m=1}^M \sum_{m \neq j}^M z^k_{u,m}(t_i) p^k_{u,m}(t_i) h^k_{u,m}(t_i) d^{-β}_{u,m}(t_i)}
\]

where \(z^k_{u,j}(t_i)\) is the RA indicator which manages the allocation of resources indicating either RB k is allocated to serve user u of mSc j in time slot t_i or not. This can be described by

\[
z^k_{u,j}(t_i) = \begin{cases} 
1, & \text{if RB } k \text{ serves user } u \text{ of mSc } j \text{ during time slot } t_i \\
0, & \text{otherwise.}
\end{cases}
\]

\(p^k_{u,j}(t_i)\) denotes the transmitting power of mSc j which is considered equal for all mScs, \(h^k_{u,j}(t_i)\) is the channel gain that follows Rayleigh distribution and evolves with time \(t_i\), \(d^{-β}_{u,m}(t_i)\) is the distance between transmitter \(j\) and receiver \(u\) in time slot \(t_i\), \(β\) represents path loss exponent, and \(N_0\) represents power spectral density (PSD) of additive white Gaussian noise (AWGN). The term \(\sum_{m=1}^M z^k_{u,m}(t_i) p^k_{u,m}(t_i) h^k_{u,m}(t_i) d^{-β}_{u,m}(t_i)\) represents interference from all other mScs to user u of mSc j over RB k in time slot \(t_i\). Where \(p^k_{u,m}(t_i)\) is the transmitting power of mSc m that is assumed to be same as other mScs in time slot \(t_i\), \(h^k_{u,m}(t_i)\) represents the channel gain following Rayleigh distribution and evolving with time \(t_i\), and \(d^{-β}_{u,m}(t_i)\) is the distance between transmitter \(m\) and receiver \(u\) over time slot \(t_i\).

Therefore, data rate expression of mSc j which is transmitting to mobile user u on resource block k during a specific time slot \(t_i\) is

\[
R^k_{u,j}(t_i) = W \log_2 \left( 1 + \gamma^k_{u,j}(t_i) \right)
\]

where \(W\) represents allocated bandwidth. The sum-rate of the access link is given as

\[
R = \sum_{i=1}^N \sum_{j=1}^M \sum_{u=1}^U \sum_{k=1}^K R^k_{u,j}(t_i).
\]

III. PROBLEM FORMULATION FOR RESOURCE ALLOCATION

This section formulates the mathematical problem of allocating K RBs to the users of the M mScs. We focus on RB allocation in non-dense as well as dense mSc network with the objective of minimizing the RB utilization while fulfilling minimum RB demands of all the mSc users. The problem formulation is given as

\[
\text{minimize} \quad \sum_{i=1}^N \left( \sum_{k=1}^K \sum_{j=1}^M \sum_{u=1}^U z^k_{u,j}(t_i) \right)
\]

subject to \(z^k_{u,j}(t_i) \leq 1 \ \forall j, k, t_i\) (5a)

\[
\sum_{u=1}^U D_j(t_i) \sum_{k=1}^K z^k_{u,j}(t_i) \geq 1 \ \forall j, t_i
\]

where \(D_j(t_i)\) is the minimum demand of mSc j, and \(z^k_{u,j}(t_i)\) is the RA indicator which manages the allocation of resources indicating either RB k is allocated to serve user u of mSc j in time slot t_i or not. This can be described by

\[
z^k_{u,j}(t_i) = \begin{cases} 
1, & \text{if RB } k \text{ serves user } u \text{ of mSc } j \text{ during time slot } t_i \\
0, & \text{otherwise.}
\end{cases}
\]

\(p^k_{u,j}(t_i)\) denotes the transmitting power of mSc j which is considered equal for all mScs, \(h^k_{u,j}(t_i)\) is the channel gain that follows Rayleigh distribution and evolves with time \(t_i\), \(d^{-β}_{u,m}(t_i)\) is the distance between transmitter \(j\) and receiver \(u\) in time slot \(t_i\), \(β\) represents path loss exponent, and \(N_0\) represents power spectral density (PSD) of additive white Gaussian noise (AWGN). The term \(\sum_{m=1}^M z^k_{u,m}(t_i) p^k_{u,m}(t_i) h^k_{u,m}(t_i) d^{-β}_{u,m}(t_i)\) represents interference from all other mScs to user u of mSc j over RB k in time slot \(t_i\). Where \(p^k_{u,m}(t_i)\) is the transmitting power of mSc m that is assumed to be same as other mScs in time slot \(t_i\), \(h^k_{u,m}(t_i)\) represents the channel gain following Rayleigh distribution and evolving with time \(t_i\), and \(d^{-β}_{u,m}(t_i)\) is the distance between transmitter \(m\) and receiver \(u\) over time slot \(t_i\).
where \( p_{u,j}^{k}(t_i) \) represents the RB allocation variable as discussed in (2). Constraint (5b) ensures that the RB assigned to a specific mSc \( j \) cannot be allotted to other mSCs that are present in its interference range over specific time slot \( t_i \). Constraint (5c) is related to the demand requirement and it ensures that the minimum RB demand of each mSC will be satisfied. Constraint (5d) and (5e) is the non-negative constraint and binary variable constraint, respectively.

It can be noticed from formulated problem that the considered minimization problem is NP-hard. The NP-hardness can be proven by mapping the considered problem to the graph coloring problem because the two problems are similar, and the graph coloring problem is proven to be NP-hard [40]. In graph coloring, we try to minimize the colors required to color the nodes of the graph. This demands different colors for any of the adjacent nodes. In our case, nodes conform to mSCs and colors conform to RBs. The adjacent nodes refer to interfering mSCs in our RA problem. Moreover, the interference part of formulated problem makes it non-convex. Consequently, RA is a challenging task because finding the optimal solution for a non-convex problem is generally intractable. Because the problem cannot be solved analytically, a PD DL-based approach is proposed. A PD DL-based mechanism is considered in this paper due to its universal capability of approximating the relationship among the given input parameters and the respective optimal solution.

IV. POSITION-DEPENDENT-LSTM RESOURCE ALLOCATION ALGORITHM

This section presents the PD-LSTM algorithm for RA where near-optimal solution can be obtained for the formulated optimization problem. In mobile dense cellular network, we consider a centralized processing mechanism in which the central hub collects the necessary information about all mSCs including the communication overheads between them on regular basis. All the training required for the resource allocation algorithm is performed offline at the BS periodically whereas the available RBs are allocated to the mSCs based on their demand for traffic in real-time.

We consider the supervised DL mechanism called LSTM for our formulated problem based on the accuracy it provides compared to KNN and ANN. The proposed algorithm uses LSTM in order to assign the resources in a mSC network by learning the relationship between inputs and outputs of a problem while treating the formulated problem as a black box. In the proposed methodology, conventional optimization mechanism is taken into consideration which will act as a supervisor in the algorithm. The output of the conventional approach will be presented as ground truth labels for LSTM. LSTM network has ability of universal approximation as it maps the input parameters to the respective outputs [37]. Fig. 3 demonstrates the training and testing phases of the proposed resource allocation mechanism. Both phases are discussed in Section IV-A and Section IV-B respectively.

A. Training Phase

Algorithm 1 presents the training phase of resource allocation using LSTM mechanism. For training the supervised network, labeled samples \( k^o \in [k^o]_{S \times 1} \) are required for the corresponding generated historic data \( \mathcal{H} \in [t^s, b^s, r^s, f^s] \). The variable \( t^s \) represents the time for a sample \( s \) with respective bus ID \( b^s \), route ID \( r^s \), and position of buses \( f^s \). The total number of samples is \( S \). The vector \( k^o \) is the vector of labels associated to the historic data \( \mathcal{H} \) and is formed by running
Calculate $y$ for $t = 1, 2, 3, \ldots , Q$

Decompose $\mathcal{X}$ in $Q$ batches, each of size $I$.

for each of the batch $q$, $q = 1, 2, 3, \ldots , Q$

$y_{i,q} = \text{LSTM}(X_{i,q})$, $\forall i$

Calculate $J(\theta_q)$ as in (6) to estimate network quality

Updating $\theta_q$ by minimizing $J(\theta_q)$ as

$\theta_{q+1} = \theta_q - \alpha \frac{\partial J(\theta_q)}{\partial \theta_q}$

Save the trained LSTM network.

the optimization algorithm which uses the historic data $\mathcal{H}$ as presented in Fig. 3. For creating the label vector $k^o$ corresponding to historic data $\mathcal{H}$, optimization algorithm involves the generation of time dependent interference graph and applying graph coloring algorithm to the generated graph [14]. Now, we have training data set $\mathcal{X}$ comprising of the mentioned historic input data vectors $\mathcal{H}$ along with their ground truths vector $k^o$, where $\mathcal{X} = [x_s, b_s, r_s, f_s, k_s^o]$. Data set $\mathcal{X}$ is taken as an input to train the LSTM network for producing the corresponding output vector $Y$, which is the vector of RBs for the respective mScs.

Once the data set is formed and input is defined, the LSTM network is trained as shown in Algorithm 1. Gradient Descent (GD) algorithm is considered while training the network because of its computational power of finding the solution. Furthermore, the training phase depends on the size of mini-batches associated to GD algorithm. Therefore, the training data set $\mathcal{X}$ is first decomposed into total of $Q$ batches and size of each batch $q$ is $I$. The later step mentioned in the algorithm is to randomly initialize the weights $\theta_q$ as well as the learning rate $\alpha$. The input data set is divided into a total of $Q$ batches and each batch has a total of $I$ samples, i.e., $S = I \times Q$, predicted RBs is computed for each batch $q$ where $q = 1, 2, 3, \ldots , Q$. Each sample of data associated, $X_{i,q} = [i_t, b_{i,q}, r_{i,q}, f_{i,q}, k_{i,q}^o]$ is provided as an input to LSTM and respective output $y_{i,q}$ is received as an output from the LSTM. The loss function for batch $q$ by determining the discrepancy present between results achieved using LSTM and the ground truth labels in order to determine the quality of trained network is computed as follows

$$J(\theta_q) = \frac{1}{2I} \left[ \sum_{i=1}^{I} (y_{i,q} - k_{i,q}^o)^2 + \lambda \sum_{i=1}^{I} \theta_{i,q}^2 \right]$$

where $y_{i,q}$ is the output of LSTM for the input $i$ of batch $q$ and $k_{i,q}^o$ represents the ground truth label that is optimal output received by the traditional method. The LSTM updates weights $\theta_q$ for each of the batch $q$ by minimizing the achieved loss function. The weights are updated with the help of GD mechanism using $\theta_{q+1} = \theta_q - \alpha \frac{\partial J(\theta_q)}{\partial \theta_q}$, where $\alpha$ represents the learning rate of LSTM. Once the network is trained for all batches of training data, trained network is saved.

B. Testing Phase

For testing phase, the sample labels are created using the same conventional approach that is used in training phase as shown in Fig. 3. Afterwards, the input vectors of testing data for time frame $T$ are given to the trained position-dependent LSTM network and corresponding outputs are collected. Then, comparison of the inferred solutions and respective ground truths are compared to evaluate the trained position-dependent LSTM.

C. Computational Complexity Analysis

1) Computational Complexity of Off-Line Processing: This sub-section provides the complexity analysis of generating the ground truth labels and Algorithm 1, i.e., the training phase.

- Complexity of Ground Truth Generation Algorithm: Graph coloring algorithm is proposed to generate the ground truth labels for training the network centrally. Hence, graph coloring method acts as a supervisor as mentioned in Section IV. The greedy graph coloring mechanism takes an interference graph and available $K$ number of RBs as an input and outputs a resource allocation matrix for $U$ users. The algorithm first organizes the nodes of a graph in descending sequence based on degree of nodes, where nodes corresponds to mSc users. Then for each RB, an RB to node allocation is performed in descending order of node degree. A total of $K \times U$ operations are performed for this. Moreover, graph coloring algorithm also ensures that the color (RB) is reutilized by other nodes (non-interfering users) which requires $U$ operations in addition to the earlier mentioned $U$ operations. Thus, the complexity of generating the ground truth labels is $O(U^2 \times K)$.

- Computational Complexity of Algorithm 1: The complexity of Algorithm 1 that is proposed to train the deep learning model for RA in mSc network is analyzed here. It can be observed in Algorithm 1, the training data set is decomposed in $Q$ batches and each of the batch $q$ trains the network with its $i$th input $(i_t, b_{i,q}, r_{i,q}, f_{i,q}, k_{i,q}^o)$ at a time. Therefore, the network takes $Q$ operations for batches and $I$ operations for each batch resulting in total of $O(Q \times I)$ operations.

Overall, the complexity of off-line processing of proposed algorithm is $O(U^2 \times K + Q \times I)$. However, the complexity of off-line processing can be reduced to $O(Q \times I)$, if the real-time RBs allocate to mScs can be acquired while collecting the real-time data of mScs.

2) Computational Complexity of Real-Time Processing: The complexity of the testing phase, i.e., the complexity of allocation of RBs to the mScs is $O(N)$, i.e., total of $N$ operations are required to predict the RBs allocated to all the mScs.
3) Computational Complexity of Overall Processing: The overall complexity of the proposed method is $O(U^2 \times K + Q \times I + N)$.

All the mScs who require resources sends the request to macro-cell BS. The proposed algorithm is centralized, information of all the mScs / mSc users is required at the central server which demands for the additional bandwidth. Therefore, there is a need of extra $E$ bytes for user per mSc resulting in overall $(U + M) \times E \times 8/N$ bits per second (bps) overhead of a mSc network [7].

V. PERFORMANCE EVALUATION

This section presents the performance of the proposed resource allocation mechanism in comparison with other machine learning and deep learning algorithms. Moreover, this section also provides the performance comparison of proposed RA mechanism with conventional RA methodologies. Data rate and number of RBs utilized in mSc network are considered as the evaluation parameters. MATLAB is used to carry out the simulations.

A. Data Generation

We considered software based generation of historic data and implement the proposed resource allocation algorithm in same software. Historic data plays a fundamental role in allocation of resources to mScs using position-dependent LSTM algorithm. Toolbox named vehicular network is used for the formation of historic data [13], [14]. The mobility characteristics of city buses involving the speed and path (route) information along with a number of other factors covering road conditions, weather, traffic peak hours, etc, impacts the city bus system. The speed and route of city buses are paid special attention while designing city bus system in a software because the city buses travel along different routes and any two buses may or may not share the same road segments resulting in dynamic interference patterns. We considered most of the factors while designing the city bus transit system in a software to make it nearly practical scenario.

Overall, the city bus system that is simulated in MATLAB is shown in Fig. 4 that is the same as [14, Fig. 7]. The modeled city bus system consists of 3 bus stops covering 2 road segments. The city bus system is modeled with 2 routes where route 1 of city buses traverses along the road segment 1 (shown using green color) and route 2 moves along the road segment 1, and 2 (shown with red color). It can be seen in Fig. 4 that route 1 and route 2 share the same road segment 1. Moreover, the velocity of city buses is also considered during the peak and regular hours of traffic while focusing on variable turning and running velocities. Fig. 5 discusses the peak hour and regular hour scheduling of the modeled city bus transit system in software and its effect on the velocity concerned with the city buses as well as its impact on utilization of resources to mScs placed in city buses. It can be seen in figure that during the regular hours of traffic, city buses travel with higher speed whereas RBs allocated to mScs equipped in city buses are lower in number because interference between mScs is less. Contrary to this, speed of city buses is reduced in peak hours of traffic and the requirement of resources increases as more mScs comes in interference range of each other. For simulation of city bus transit system, we considered the small portion of peak hour and regular hour scenarios mentioned in Fig. 5. The varying velocities are considered in both scenarios to simulate the city bus system according to the defined regular and peak hours of traffic as well as for the turning points. City buses move with variable speed of 16-18 meters/sec during peak hours of traffic and 20-22 meters/sec in regular hours of traffic. Moreover, the turning velocities of city buses are also defined as 3-4 meters/sec during peak hours of traffic and 6-7 meters/sec during regular hours of traffic. Two types of environments are setup for generation of data as follows.

1) Non-Dense City Bus Environment: In first city bus system, total of 8 city buses are considered traveling along two routes, from which, 4 buses follow one route and other 4 buses travel on the other route. The headway characteristics of one minute between the city buses of route 1 as well as between the buses of route 2 are considered while designing the non-dense city bus system.

2) Dense City Bus Environment: In second setup, city bus system comprises of total of 30 city buses following two routes. 15 buses travel along route one and 15 follows the second route in this environment setup. The
headway of 30 seconds between the city buses and the headway of 30 seconds between the buses of route 2 are taken into account while setting up the dense city bus environment.

Overall, the northbound direction of city buses is considered to generate the historic data comprising of time instants, bus IDs, route IDs and bus positions, for four days. The data is generated for every second of total of 15 minutes per day, from which, 10 minutes belong to peak hours traffic and rest of the 5 minutes come under the regular hours of traffic. Table II summarizes the specifications for modeling both the mentioned city bus systems in software.

**B. Implementation of Deep Learning-Based Resource Allocation**

The generated data is divided into training data and testing data as 3:1 (i.e., 3 days data as training data and 1 day data as testing data). LSTM network is used for allocation of RBs to mScs in a mSc network. This section provides both the training phase and the testing phase.

1) **Training Phase:** To train the PD-LSTM algorithm for RB allocation in mSc networks, the generation of ground truth labels is required. Towards this goal, we considered the approach proposed in [13] for determination of interference patterns between mScs at each time instant to form the Time Interval Dependent Interference Graph (TIDIG). The historic data is used to find the interference relationship among mScs. Afterwards, the graph coloring mechanism is considered to allocate RBs to mScs forming TIDIG while considering the minimum demands of mobile users. The generation of ground truth labels is performed for 15 minutes a day considering all the road segments associated with the city bus system. The channel conditions are considered as quasi static for one second because the vehicles travel with maximum of 22 meters/sec and 18 meters/sec during regular and peak hours, respectively. We assumed that each mobile user requires one RB and a single mobile user is connected to each of the mSc. As there exists total of 8 city buses in first city bus network and each city bus is equipped with one mSc, so we have total of 8 mScs in our first system model forming a non-dense mSc network. Correspondingly, we have 30 city buses in our second city bus environment, in which, each bus is equipped with single mSc resulting in total of 30 mScs establishing a dense mSc network. Simulations for allocation of RBs (generation of ground truth labels) is done for total of 15 minutes in a day. We considered maximum power of 19mW associated to each mSc because there exists larger distance between mScs, and the noise power spectral density is $8.1 \times 10^{-16}$ W/Hz. Once the resources are allocated, we use these as ground truth labels in proposed algorithm. Therefore, we have the input historic data and their corresponding labels for both the mentioned cases that are required to train the LSTM network. The training process continues until the error between LSTM predicted output and the optimized output is reduced below the specified threshold, where the error reduction is achieved by adjusting the weights of the network. It is worth noting that the entire training process of the network is performed offline, and thus, it is independent of the computational time required for the RA process. The architecture of LSTM used to predict the allocation of RBs in both the considered environments is presented in Fig. 6. The LSTM network for non-dense network is trained using one hidden layer comprising of total of 50 neurons as presented in Fig. 6(a). Furthermore, the prediction accuracy of a network depends on the epoch size and mini-batch size, we have used total of 80 epochs and mini-batch size of 16 in our non-dense network scenario. Contrary to that, LSTM is trained using one hidden layer composed of 80 neurons for a dense mSc network as provided in Fig. 6(b). However, LSTM network for dense environment is trained at its best with same number of epochs as of non-dense network and mini-batch size of 64. Gradient Descent is considered as optimization algorithm and regularization is used as loss function while training the LSTM network for non-dense mSc network as well as for dense mSc network.

2) **Testing Phase:** The testing of the trained position-dependent LSTM of non-dense and dense mSc network is done using testing data which results in maximum achievable accuracy of 98.2% and 97.1259% respectively.

Fig. 7 provides the comparison of convergence plots of existing optimization algorithms of the proposed LSTM-based RA mechanism for both the non-dense mSc network and the dense mSc network. It can be observed that for both the scenarios, the convergence speed of gradient descent algorithm is significantly faster as compared to adam. Combining the loss comparison of both scenarios, it can be observed that LSTM with optimization method Stochastic Gradient Descent with Momentum (SGDM) performs better in comparison with adam. Besides, LSTM using the SGDM has better prediction accuracy in comparison with adam. Moreover, optimization algorithm named RMSProp is not considered in convergence comparison because accuracy reduces significantly when RMSProp is used as an optimization algorithm for RA in mSc network.
The proposed algorithm is considered for RA in mSc network because in [33], it is presented that [14] is the only work that incorporated LSTM as a sub-step in RA to mSc network. Moreover, it outperforms other machine learning (ML) and DL-based algorithms. For comparison of proposed algorithm with ML and DL-based methods, we considered k-Nearest Neighbor (kNN) classification algorithm and the Artificial Neural Network (ANN). Table III presents the maximum accuracy achieved using the mentioned algorithms in both the non-dense mSc network and dense mSc network. Overall, for both the discussed mSc network environments, position-dependent LSTM RB allocation algorithm outperforms ANN and kNN algorithms in terms of accuracy. On the other hand, kNN provides the lowest accuracy in allocation of resources. Moreover, the accuracy of all the discussed algorithms decreases with the increase in mScs density.

To begin, the accuracy achieved using position-dependent LSTM RA in non-dense mSc network fell from 98.2% to 97.1% in dense mSc network. Similarly, the maximum achievable accuracy of RA using kNN dropped 4.1% in dense mSc environment as compared to non-dense. However, RA in case of ANN fell dramatically from 94.1% to 84.6% when the density of mScs increases.

Furthermore, we have determined the average data rate of an mSc user achieved via DL and ML-based RA algorithms. Fig. 8 provides relationship among the interference range of mScs and average data rate achieved per user in mSc network using kNN-based RA, ANN-based RA to mScs, and position-dependent LSTM-based RB allocation to mScs. Overall, the average achievable data rate has shown a decreasing trend with the increase in interference range of mScs. This happens because mispredictions of RBs allocated to mScs increases...
as the interference range rises depending on the nature of algorithm. Moreover, for both the considered mSc network scenarios, position-dependent LSTM RA has accounted for considerably improved data rate per user in comparison with RA using kNN and ANN. Furthermore, the reduction in data rate is seen using all the mentioned algorithms when the density of mScs increases.

Regarding Fig. 8(a), the average achievable data rate per user in non-dense mSc environment, it is notable that average data rate attained by position-dependent LSTM RA is 7.7% and 15.1% more as compared to data rate achieved by ANN-based and kNN-based RB allocation respectively. Moreover, data rate attained by employing ANN-based RA is 7.9% better in comparison with RA using kNN because we have also seen in Table III that ANN-based RA provides 7.56% more accurate allocation of RBs to mScs as compared to RA using kNN. Contrary to that, data rate achieved by RA using position-dependent LSTM model in dense mScs network is 19.3% and 27.4% better in comparison with data rate achieved using ANN and kNN-based allocation of resources as shown in Fig. 8(b). Apart from that, data rate achieved per user in non-dense mSc scenario using position-dependent LSTM RA algorithm, RA using ANN, and allocation of resources in kNN declined in dense mSc environment.

Therefore, it is clear that resource allocation using position-dependent LSTM algorithm yields improved results in terms of prediction accuracy, and achievable data rate as compared to other machine learning and deep learning-based resource allocation mechanisms.

Afterwards, we have compared the results of proposed method with existing resource allocation methods. Fig. 9 provides the relationship among mScs interference range and number of RBs allocated to mScs with the help of bar graph. The bar graph shows an upsurge in RB utilization in mSc network using TIDIG-based RA mechanism [7], RA using TIPDIG [14], Global Positioning Dependent Interference Graph(GPS-DIG)-based allocation of resources, as well as the proposed position-dependent LSTM RA methodology.
Overall, it is to be noted that TIDIG graph-based allocation of resources utilized the maximum number of RBs as it allocates the optimal number of RBs to mScs in both the non-dense and dense mSc cases. Whereas, RBs used by the proposed mechanism is slightly less than TIDIG-based RB allocation but comparable to the ones used by Threshold Percentage Dependent Interference Graph (TIPDIG)-based RA. Moreover, proposed mechanism allocated more RBs in comparison with the RBs allocated using GPS-DIG. Apart from that, it is also observed that the number of RBs utilized in dense mSc network is more in comparison with RB required in non-dense mSc network.

Fig. 10 represents the relation between interference range of mSc and data rate achieved per user in non-dense and dense mSc scenarios using different RB allocation schemes. In general, considering both non-dense and dense mSc networks, it is shown in figure that data rate achieved using TIDIG-based RA stays relatively equal over various measures of interference range. This is due to interference free nature of TIDIG where one RB is allocated to each individual mSc while diminishing the interference present between mScs. On the other hand, average data rate attained per user by employing GPS-DIG-based RB allocation, and RA using TIPDIG shows the deteriorating affect as interference range of mScs increases. This is due to non-consideration of some of interference relationships present between mScs. Similarly, proposed position-dependent LSTM RA also shows declination in achievable data rate with rise in interference range because of increase in mispredictions. GPS-DIG-based allocation of RBs just considers the interference patterns between mScs present at starting instant of time period and uses the same interference relationships for whole time frame. Similarly, in TIPDIG-based RA, percentage threshold is considered to diminish the effects of insignificant interference patterns (interference between mScs that occurs less than 5% of time frame) while allocating RBs to mScs which simultaneously avoids interference to greater extent with escalation in interference range. Moreover, the proposed position-dependent LSTM RA mechanism mispredicts some of the interference relationships present within mScs. Furthermore, deterioration in data rate per user is observed in all the discussed mechanisms when the number of mScs increases in network.

In non-dense mSc scenario, it is noticed that data rate attained by applying TIDIG-based RA is 23%, 17.4%, and 2.8% more as compared to GPS-DIG, LSTM, and TIPDIG-based RA algorithms respectively as shown in Fig. 10(a). Also, the data rate attained by proposed position-dependent LSTM RA is 12.4% less in comparison with TIPDIG-based RA and 4.77% more with respect to RA using GPS-DIG. On the other hand, in dense mSc network, RA using TIDIG achieved 55.4%, 39.2%, and 6.1% higher data rate in comparison with GPS-DIG-based RB allocation, RA using position-dependent LSTM, and RB allocation using TIPDIG respectively as presented in Fig. 10(b). In addition to that, RB allocation using position-dependent LSTM algorithm achieved 54.2% less data rate as compared to TIPDIG-based mechanism and 26.7% more data rate per user in comparison with GPS-DIG-based RA.

The computational time is one of the main challenges for adopting deep learning in real-time applications. Therefore, the computational time of the proposed PD-LSTM is evaluated and compared to other well-established methods as depicted in Table IV. It is worth noting that the computational time in such scenarios does not include the training time, because it is typically performed off-line, and hence it does not affect the suitability of the proposed algorithm for real-time applications. The table presents the computational time for one interference range of 200 meters, which corresponds to a single optimization process, and the total time for six equally spaced interference ranges between 50 and 300 meters. The relative CPU time is computed as the ratio of time required for the dense network. As can be noted from the table, the proposed PD-LSTM significantly outperforms the TIDIG and TIPDIG algorithms, while the GPS-DIG outperforms the PD-STM by a factor of two, but that comes at the expense of rate reduction.
Although the computational time for a single allocation process is a fraction of a second, which is relatively a short time, such time might be considered long for vehicular applications where the channel coherence-time is typically in the order of tens of milliseconds in certain scenarios [41]. However, it should be noted that the results presented in Table IV are obtained using a general purpose computing machine, and thus, the computational time would be substantially less if the algorithm was evaluated using a dedicated and optimized hardware. In the proposed framework, the optimization process is performed at the BS, and thus, edge server-based architectures can be adopted to provide fast solutions for real-time applications [42]. Edge computing has already been adopted for such applications. For example, a major cellular Internet service provider in the United States and a national fast-food chain have deployed edge computing services [42]. Hybrid edge-cloud solutions can also be adopted given that the edge node and cloud server are generally close to each other to avoid the routing delay as reported in [42] and the references listed therein.

Overall, it is noted that proposed algorithm provides improved results in comparison with some machine learning algorithms in terms of accuracy and average achievable data rates. Moreover, the computational speed comparison reveals that the proposed strategy performs better than existing TIDIG and TIPDIG-based mechanism whereas it provides lesser data rate in comparison with them. Similarly, position-dependent LSTM RA is computationally expensive in comparison with GPS-DIG-based RA mechanism but provides better data rates as compared to existing RA using GPS-DIG. Although the existing TIDIG and TIPDIG-based allocation of RBs provides noteworthy data rates for the computed results, it is almost infeasible for them to handle the users requirements in real-time as shown in computational speed comparison.

VI. CONCLUSION AND FUTURE WORK

This paper examined DL-based RB allocation to mScs integrated in city buses to serve the mobile users in non-dense mSc network and dense mSc network. LSTM neural networks are considered for RB allocation which demands for historic data and their respective labels. The labels are generated using optimization method that act as supervisor in our algorithm. The network is trained by diminishing the error present between predicted RBs allocated to mScs and optimized RBs generated by the optimization algorithm. We have seen that position-dependent LSTM RA provides better results than other machine and deep learning-based RA methods in both the considered scenarios. Moreover, in non-dense mSc case and dense mSc environment, it is noticed that data rate attained using the proposed algorithm is less in comparison with TIDIG-based RB allocation and TPDIG-based allocation of RBs. But, proposed methodology performs better than GPS-DIG-based RA. However, data rate reduces for all the algorithms except TIDIG in dense mSc network in comparison with non-dense environment. It is also seen that position-dependent LSTM-based RA is computationally less expensive than TIDIG-based and TIPDIG-based RA algorithm. Proposed methodology is also easy to implement and capable to deal with real time scenarios as compared to other resource allocation mechanisms. In future, distributed learning approaches can be considered along with data generated from realistic city bus transit system.

REFERENCES

[1] “World telecommunication development report 1999, mobile cellular,” Int. Telecommun. Union, Geneva, Switzerland, Rep., Oct. 1999. [Online]. Available: https://www.itu.int/ITU-D/ict/publications/wtdr_99/material/wtdr99s.pdf
[2] “Mobility Report,” Ericsson. 2019. [Online]. Available: https://www.ericsson.com/4ac7e7e/assets/local/mobility-report/documents/2019/emr-november-2019.pdf
[3] “Cisco Annual Internet Report,” Cisco. 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html
[4] A. Shahid, V. Maglogiannis, I. Ahmed, K. S. Kim, E. De Poorter, and I. Moerman, “Energy-efficient resource allocation for ultra-dense licensed and unlicensed dual-access small cell networks,” IEEE Trans. Mobile Comput., vol. 20, no. 3, pp. 983–1000, Mar. 2021.
[5] J. Rygaard, “Digitization, densification and why you should care,” Nokia, Espoo, Finland, Rep., Feb. 2017. [Online]. Available: https://www.nokia.com/blog/digitization-densification-care/
[6] J. Qiu et al., “Hierarchical resource allocation framework for hyper-dense small cell networks,” IEEE Access, vol. 4, pp. 8657–8669, 2016.
[7] S. Jangsher and V. O. K. Li, “Resource allocation in moving small cell network,” IEEE Trans. Wireless Commun., vol. 15, no. 7, pp. 4559–4570, Jul. 2016.
[8] S. Basnet, Y. He, E. Dutkiewicz, and B. A. Jayawickrama, “Resource allocation in moving and fixed general authorized access users in spectrum access system,” IEEE Access, vol. 7, pp. 107863–107873, 2019.
[9] Z. Iftikhar, S. Jangsher, H. K. Qureshi, and M. Aloqaily, “Resource efficient allocation and RRH placement for backhaul of moving small cells,” IEEE Access, vol. 7, pp. 47379–47389, 2019.
[10] M. Kalil, A. Shami, and A. Al-Dweik, “QoS-aware power-efficient scheduler for LTE uplink,” IEEE Trans. Mobile Comput., vol. 14, no. 8, pp. 1672–1685, Aug. 2015.
[11] J. Qiu, Q. Wu, Y. Xu, Y. Sun, and D. Wu, “Demand-aware resource allocation for ultra-dense small cell networks: An interference-separation clustering-based solution,” Trans. Emerg. Telecommun. Technol., vol. 27, no. 8, pp. 1071–1086, 2016.
[12] H. Chen, D. Zhao, Q. Chen, and R. Chai, “Joint computation offloading and radio resource allocations in small-cell wireless cellular networks,” IEEE Trans. Green Commun. Netw., vol. 4, no. 3, pp. 745–758, Sep. 2020.
[13] S. Zafar, S. Jangsher, M. Aloqaily, O. Bouachir, and J. B. Othman, “Resource allocation in moving small cell network using deep learning based interference determination,” in Proc. IEEE PIMRC, Istanbul, Turkey, 2019, pp. 1–6.
