Journal of Network and Computer Applications 212 (2023) 103567

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Sequence-to-sequence learning for link-scheduling in D2D communication networks

Ahmed Elsheikh [a,*], Ahmed S. Ibrahim [b], Mahmoud H. Ismail [c,d]

[a] Cairo University, Mathematics and Engineering Physics Dept., Cairo University, Giza, 12613, Egypt
[b] Florida International University, Electrical and Computer Engineering, Dept., Miami, 33174, FL, USA
[c] American University of Sharjah, Electrical Engineering Dept., Sharjah, 26666, United Arab Emirates
[d] Cairo University, Electronics and Communications Engineering Dept., Giza, 12613, Egypt

ARTICLE INFO

Handling Editor: Dr. M. Atiquzzaman

Keywords:
Device-to-Device
Machine learning
Scheduling
Sequence-to-Sequence

ABSTRACT

Scheduling of Device-to-Device (D2D) links in communication networks conventionally relies on solving NP-hard combinatorial optimization problems. These types of solution approaches will not be suitable for the service requirements of future networks due to the associated computational complexity. That is why Deep Learning (DL) is one of the promising approaches to tackle this problem. Nevertheless, designing the learning algorithm to cope with the dynamic nature of the D2D network is a challenge. Current research using DL only assumes a static layout of the network without taking advantage of the correlations between the decisions in a dynamic network. Consequently, this paper proposes a sequence-to-sequence modelling (SSM) method for D2D scheduling using only distance information. The SSM processes the distance information as well as the previous scheduling decisions in a sequential manner with a feedback from the intermediate output, and models the correlations between consecutive input information as well as the produced decisions. Simulation results show that the average sum rate of the SSM is about 95% of that achieved by the optimal scheduler and it requires at least 90% less resources than those required by other DL schedulers reported in the literature. Finally, the decision-making of SSM is explored for key input descriptors and an unsupervised decision-maker is explored, which is shown to produce reasonable results with minimal computational requirements.

1. Introduction

Devices and their applications have evolved significantly over the recent years. Clearly, such applications are service demanding, but future networks will need to take advantage of the availability and abundance of distributed devices to decrease the centralized load as much as possible. Hence, different communication modes are envisioned to co-exist within the same network (Soldani et al., 2014). Among these modes is Device-to-Device (D2D) communication, which refers to direct communication between user equipment. One of the main challenges facing D2D networks is resource management, particularly the scheduling of active links. Generally, a base station needs to quickly assign resources and identify transmission powers to devices coming into and staying within its coverage area, including those in D2D communication modes.

A lot of work on solving the D2D link scheduling can be found in literature using fractional programming such as FLashLinQ, ITLinQ, and FPLinQ (Shen, 2020). Nevertheless, to cope with the demanding modern wireless services, many works in the literature such as (Ye et al., 2019), (Luo, 2020; Chen et al., 2019; Cui et al., 2019) sought solutions using Deep Learning (DL) algorithms to avoid solving cumbersome optimization problems at the base stations. More precisely, various DL models have emerged as effective approaches to achieve multiple goals in D2D networks including power allocation, spectrum sharing, communication mode selection, and efficient resource allocation while requiring lesser computations and estimations (Luo, 2020), (Lee et al., 2020). Spatial learning proposed by (Ba and Caruana, 2014) and graph embedding proposed by (Prabhavalkar et al., 2017) and (Shen et al., 2020) are two significant examples of such attempts. Both approaches try to define neural network (NN) structures that automatically encode distance information. The former extracts interference and distance information based on kernel filters that are learned from synthetically generated data. However, in order to learn a sufficient number of filters to solve the scheduling problem, a significant amount of training data is required.

* Corresponding author.
E-mail address: ahmed.elsheikh@eng.cu.edu.eg (A. Elsheikh).

https://doi.org/10.1016/j.jnca.2022.103567
Received 31 December 2021; Received in revised form 21 October 2022; Accepted 20 December 2022
Available online 30 December 2022

On the other hand, the graph embedding approach eases the burden of learning the mappings through kernel filters by preparing neighborhood graphs describing the network through pairwise distances. What is then required to learn is the embedding from graph to vector representation and feed it to the network since the NN can only work with vectors. Although the spatio-temporal variations in communication links are intrinsic to dynamic networks as a result of mobility as well as the changes in the environment, neither of the two approaches mentioned above takes into consideration the correlation between the scheduling decisions neither in time nor in space.

It is worth mentioning here that the modelling capability of a DL model is highly dependent on its structure and how the information is processed within. For D2D networks, users' mobility and relative locations are correlated to one another and the scheduling decisions taken for each pair are also correlated. In addition to the channel variations for a given device, the number of devices inside a certain coverage area will also change, which will consequently affect the distribution of the resources among the serviced devices. In other words, a base station needs to quickly respond to a varying number of D2D pairs inside its coverage area. This inter-dependency requires a model with internal feedback to capture such complex correlations. Moreover, these inter-dependencies do not generally follow a known pattern.

Based on the above discussion and inspired by the successful employment of sequence-to-sequence modelling (SSM) in dealing with correlated data in speech recognition and language processing where data is treated sequentially to produce a sequential set of correlated outputs (Haykin, 2010), this paper proposes the adoption of SSM for D2D link scheduling. The idea is to represent each D2D pair with a set of descriptors that rely solely on the devices' locations, which can be acquired using global positioning system (GPS), for example, without the need for the fast-varying small-scale fading information (i.e., instantaneous channel state information (CSI)). In dense urban networks where acquiring GPS data might be challenging, other techniques might be used as well. This includes using the base stations in the cellular network as anchor nodes to locate the users as in (Liu et al., 2021), using LiDAR technology (Yu et al., 2021), using images from Google Street view (Salarian, 2015) and using ultra-wideband ranging (Jourdan et al., 2005), just to name a few. The usage of SSM has many advantages. First, SSM inherently captures relations between the inputs without requiring external structural modifications such as using multiple kernels as in spatial learning or intermediate calculations as in graph embedding. In the same time, SSM is able to generalize to different network layouts since there are no restrictions on the number of input pairs. Moreover, SSM is able to achieve reduction in the NN model size by modelling the spatial correlations, which are inherited from the interference, and the temporal correlations inherited from mobility in dynamic networks. In addition, it achieves higher decision-making speeds and adaptability to network layout variations. Furthermore, it can handle different numbers of D2D pairs without requiring modifications since it models variable-length sequences with no constraint on the number of inputs. Although previous works using graph embedding, like (Shen et al., 2020), had the capacity to handle a variable number of D2D pairs, it was only without any mobility assumptions. In addition, this was achieved through an embedding of the network connected graph, but not from a direct modelling approach such as in the case of SSM. This adds more computational requirements to the model as will be carefully detailed in the sequel. Moreover, SSM has a built-in sequential inter-dependency between the inputs and outputs that relaxes the requirement of adding extra processing neurons in the used NN. This type of modelling was not considered in the previous papers, and to the best of the authors' knowledge, this is the first attempt to use SSM for link-scheduling in such scenario.

The paper also analyzes the results of decision-making and proposes an unsupervised scheduling scheme based on the observations. This goes in line with the attempts discussed in (Cui and Yu, 2020) where the goal is to decrease the reliance on labelled data, and to decrease the computations required to achieve acceptable performance. The main contributions of this paper can thus be summarized as follows.

- Employing a learning scheme that can concurrently model the spatiotemporal correlation in D2D link scheduling.
- Selecting a set of informative descriptors for each D2D pair to achieve high performance without requiring any learned mapping, which leads to a significant reduction in the computational requirements.
- Achieving comparable performance with much less computational and storage requirements relative to previous DL techniques for D2D link scheduling with a quantitative comparison based on statistical tests.
- Exploring the usage of key descriptor factors learned by the SSM to train an unsupervised model to achieve acceptable decision-making with minimal computational burden using a simple threshold.

The work herein also considers two simulation scenarios: the quasi-static case where no correlation is assumed and is considered for comparison with previous works in the literature, and the dynamic case where mobility is assumed. The rest of the paper is organized as follows: in Section 2, we present the system model and formulate the link-scheduling problem for a network.

Of D2D pairs. In Section 3, we describe the proposed SSM method to solve the space-time problem. In Section 4, we present the quasi-static simulation results followed by the dynamic results. The unsupervised decision-making results are then discussed in Section 5, before the paper is finally concluded in Section 6.

2. System model

We consider a network with N_d D2D pairs that are assumed to be located randomly in a two-dimensional square area with length L and the separation distances between the D2D pairs are uniformly distributed between $l_{\min}$ and $l_{\max}$. A graphical representation of the network layout is shown in Fig. 1 with several D2D pairs. The D2D pairs are assumed to move in random directions while keeping the separating distance between the pair members constant. This can be a valid assumption in multiple scenarios like in the case of vehicle-to-vehicle (V2V) communication where vehicles will be moving with constant speeds in traffic or in the case of groups of users sharing content while walking in a mall together, for example. Furthermore, we assume the

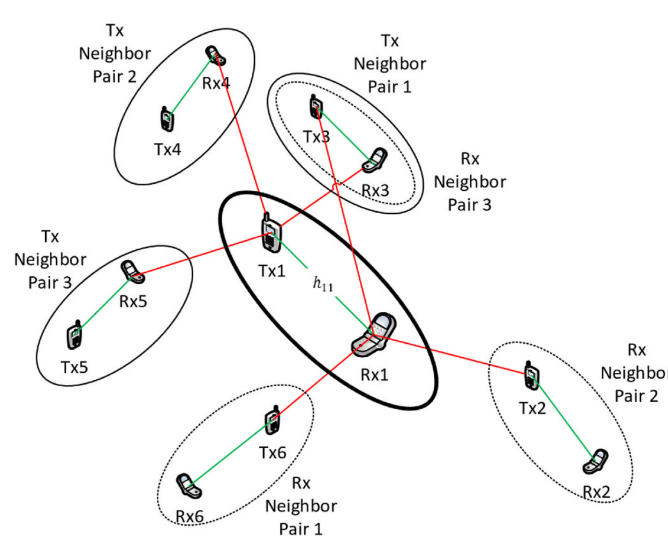

Fig. 1. Channel links between transmitters and receivers in a given D2D network. Each ellipse shows a pair of devices involved in a D2D communication session.

devices' transmission powers to be constant and we denote the channel gain for the communication link between the transmitter (Tx) device of the lth D2D pair and the receiver (Rx) device of the kth pair by h_{lk} where $l,k \in \{1,2, \ldots, N_D\}$. The goal of optimal D2D scheduling is to identify which of the links need to be active to maximize a certain performance metric. Typically, the sum rate of the whole network is chosen as a performance metric, where the rate for the link between the Tx and Rx of pair l is defined by.

$$R_l = B \times log_2\left(1 + \frac{|h_{ll}|^2 P_l \delta_l}{\sum_{k \neq l} |h_{kl}|^2 P_k \delta_k + \sigma_n^2}\right) \quad (1)$$

In the above equation, B is the link bandwidth, P_l is the power of the Tx of the lth link, σ_N^2 denotes the additive white Gaussian noise power and $\delta_k \in \{0,1\}$ indicates whether the kth link is active or not. The channel gain in (1) captures two different effects, namely the path loss (distance-dependent losses) and the shadowing effect, which is modeled using a lognormal random variable with zero mean and standard deviation σ_s.

Although distances do not show up explicitly in Eq. (1), they indirectly affect the achievable rate through the value of the channel gain.

Based on the above formulation, optimal scheduling targets finding the optimal values of δ_k for maximum sum rate for the N_d devices in the network as follows:

$$\max_{\delta_k} \sum_{k=1}^{N_d} \delta_k R_k,$$

$$\text{subject to} \sum_{k=1}^{N_d} \delta_k \leq N_d$$

which is a typical NP-hard optimization problem that requires CSI collection at a central node from each of the links.

3. SSM for D2D space-time link scheduling

In this work, SSM is proposed to process a series of input vector descriptions that use only distance information about each D2D pair to produce a series of correlated scheduling decisions. These vectors will be described in detail in Section 3.1. The model then learns from the optimal scheduler FPLinQ (Shen and Yu, 2017) its binary scheduling decisions and acts as an implicit channel estimator given a certain network layout and previously seen link-scheduling decisions. The model takes advantage of the spatial correlation between each D2D pair and its neighboring pairs, which is the result of the interference channels. The interference perceived by the D2D pair under study (indicated by the bold ellipse in Fig. 1) is dependent on the layout of its neighborhood as clearly shown in the figure. There is also correlation through time, which is a result of the D2D mobility and for typical pedestrian speeds, it is expected that there will be a high correlation between the consecutive scheduling decisions. The change in location relative to time is shown in Fig. 2, where the vector descriptions of the pairs of transmitters and receivers can be processed based on the change in location starting from the first time they enter the coverage area. SSM typically relies on the well-known recurrent neural network (RNN) architecture as will be explained in the sequel.

The first step in SSM is to describe the network layout upon which scheduling decisions will be made as detailed in the following subsections.

3.1. The proposed SSM inputs

As mentioned before, the status of each D2D pair will be represented by a set of descriptors. The proposed architecture uses only Euclidean distance information between the different devices as the descriptors,

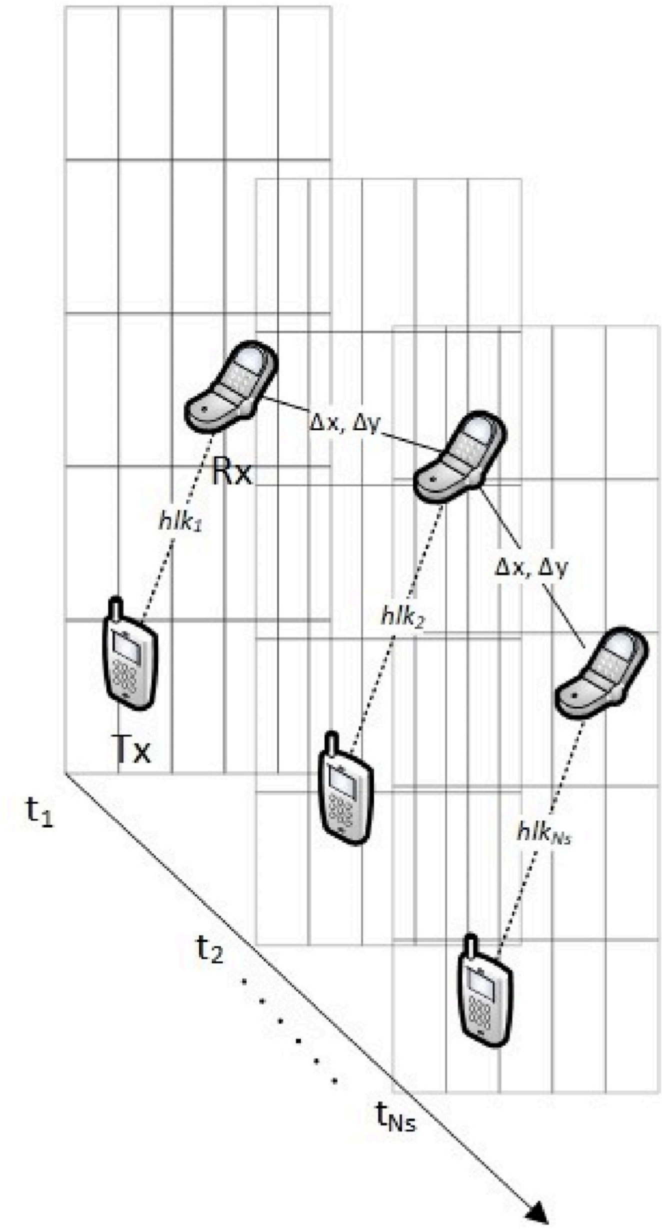

Fig. 2. Channel links between a given D2D pair as they change location through time.

which can be easily acquired using GPS or otherwise, as mentioned earlier. Fig. 1 shows the distances used as local descriptors extracted for each D2D pair as colored lines. The red lines indicate interference and the green ones indicate intended communication links. The number of features representing each D2D pair is thus $d = 4K + 1$, where K, which is a design parameter, is the number of nearest neighbors. Those neighbors are defined as the closest K Txs to the Rx of the pair under study, as well as the closest K Rxs to the Tx of the pair under study. This results in $2K$ distances, which are used as descriptors, the remaining $2K + 1$ descriptors are the separating distance between those neighbors and their other end of communication, as well as the distance separating the pair under study. The value of this parameter is chosen by cross-validation as it is not learned from the data. The cross validation process is done by sub-sampling the training data into smaller training and validation groups. By using different groups for training and validation, an unbiased estimate of the hyper-parameters such as K can be obtained.

Fig. 1 highlights the special case with $K = 3$ where the D2D pair under study is encompassed by the thick ellipse. In that case, there will be $d = 13$ descriptor values as follows: first, the interference distances from Tx1 of the D2D pair under study to the K neighboring receivers and from the K neighboring Txs to Rx1 of the pair under study are extracted, so this accounts for $2K = 6$ descriptors, 3 in each set. These are indicated by the red lines connecting Tx1 to its 3-nearest neighboring receivers and those connecting Rx1 to its 3-nearest neighboring Txs in Fig. 1. Next, we extract the separating distances between the Txs and Rxs of the 3-nearest neighbors to both Tx1 and Rx1, those are encompassed by solid ellipses for the Tx neighbors, and by dashed ellipses for the Rx neighbors, so that is another $2K = 6$ descriptors and those are highlighted in green in the figure. Finally, we include the separating distance between Tx1 and Rx1, and that constitutes the remaining descriptor, also highlighted in green in the figure.

3.2. SSM proposed architecture

The SSM is a variant of RNNs, which are a special type of NNs that are used to handle a series of inputs. In literature, the series can be a time series as in the case of speech recognition, a spatial series as in the case of optical character recognition (OCR) or a contextual one as in translation (Hochreiter and Schmidhuber, 1997). The main idea is feeding the output of the hidden neurons back to the input with some transformation weights. Fig. 3 illustrates the structure of the recurrent neuron in compact and unrolled forms over the series where w_1 through w_d are the components of the weight matrix that is multiplied with each component of the input vector $\in \mathrm{R}^d$ and then a non-linear function is applied.

Another way to interpret the RNN in terms of the scheduling problem is to view each neuron as a programmable logic gate, which is activated whenever there is a certain relationship between the distances of the pair under study. Each neuron, after training, becomes responsible for identifying a certain relationship between the current descriptors and its previous output. In the end, the aggregation of the decisions of the different neurons by a second level of neurons produces the scheduling decisions at each time step as required. This process is what is commonly known in the literature as weight sharing through time (Goodfellow et al., 2016).

Fig. 4 illustrates the proposed SSM architecture showing how an SSM processes the input sequence of D2D descriptors. As illustrated in Sec. 3.1, the descriptors define the local neighborhood of a given D2D pair with K neighbors as detailed earlier. The quasi-static sequence is processed as the series of D2D pair descriptors are fed one-by-one to the SSM for a given layout. The starting point of the series is an arbitrary pair and then the pairs are scanned going from one pair to the next closest one, which is determined through the Euclidean distance between the pairs calculated between their midpoints. Hence, each training instance is a series describing the D2D pairs spatial relationships with each other as they are laid out. On the other hand, in the dynamic case, each D2D pair is observed over time. Each time step is a descriptor describing the local neighborhood information of each D2D pair and hence, the series of which describes how the local network changes over time and space. Accordingly, each training instance in the dynamic case is a series of spatial information about a certain D2D pair.

In the quasi-static case, each input descriptor vector for each D2D pair is processed by a set of Gated-Recurrent Units (GRUs) depicted as the blue circles in the figure, which are a more stable variant of the regular RNN. The descriptors are taken in sequence depending on the location of the D2D pair. A feedforward NN is then used to produce a binary decision of whether the link is active or not. The ability to work with sequences allows the SSM to mitigate any restrictions on the number of D2D pairs because the recurrent units in the SSM structure allow them to process a series of any size without requiring to be retrained, nor do they require new output units to be added or removed to accommodate changes in the number of D2D pairs. For the dynamic case, on the other hand, each D2D pair is considered to have a sequence of descriptors that change over time. Accordingly, the SSM processes these sequences and produces the results of each time step for each D2D pair. The space dependencies are implied in the descriptors, which are used to train the SSM with different network layouts.

3.2.1. Training the SSM

The SSM is a supervised learning model that requires target outputs to achieve the desired task. The data used for our model is composed of the input vectors as described in Section 3.1 and the outputs are the scheduling decisions made by the optimal scheduler FPLinQ (Shen and Yu, 2017) with a specific limit set on the maximum number of iterations. The learning algorithm is backpropagation through time (Goodfellow et al., 2016) using the Adam optimizer (Jozefowicz et al., 2015) and the loss function adopted is the binary cross-entropy (De Boer et al., 2005). The weight updates are done through the optimizer using the D2D scheduling training dataset, where each observation in the dataset is a series of locations and neighborhood information and its corresponding scheduling decision.

3.2.2. Processing neurons

A variant of the classical RNN neurons, which is now a common building unit for series NN, is the Long-Short Term Memory (LSTM) cell. LSTM cells mitigate the problem of vanishing and exploding gradients, which is faced in training classical RNNs (Friedman and Hastie, 2009). A proposed simplification of the LSTM cell is the GRU in order to decrease the number of trainable parameters while maintaining comparable performance (Jozefowicz et al., 2015). The GRU has three control signals; each is a non-linear function activated by a weighed sum of the current input observation $\mathbf{x_j}$, which can be one of the descriptors

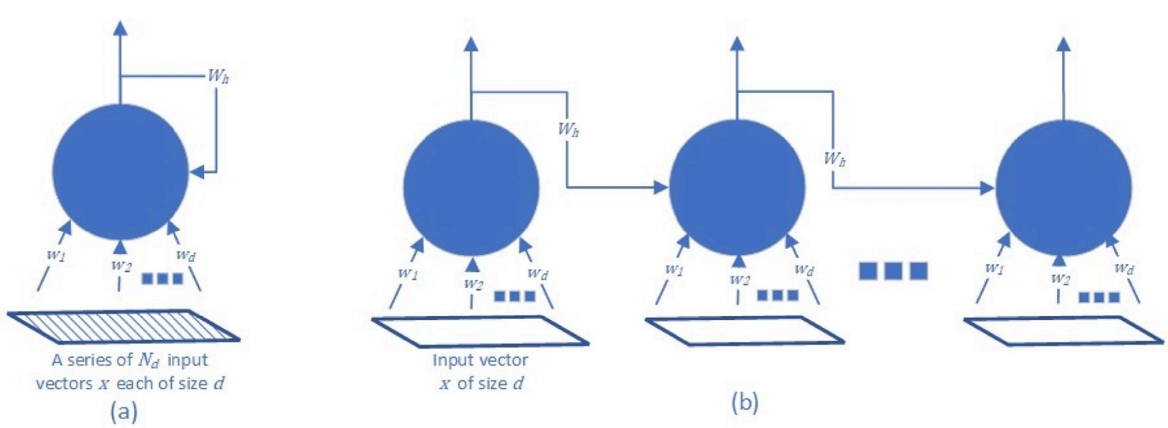

Fig. 3. Recurrent neuron connections: (a) compact (b) unrolled over an input series.

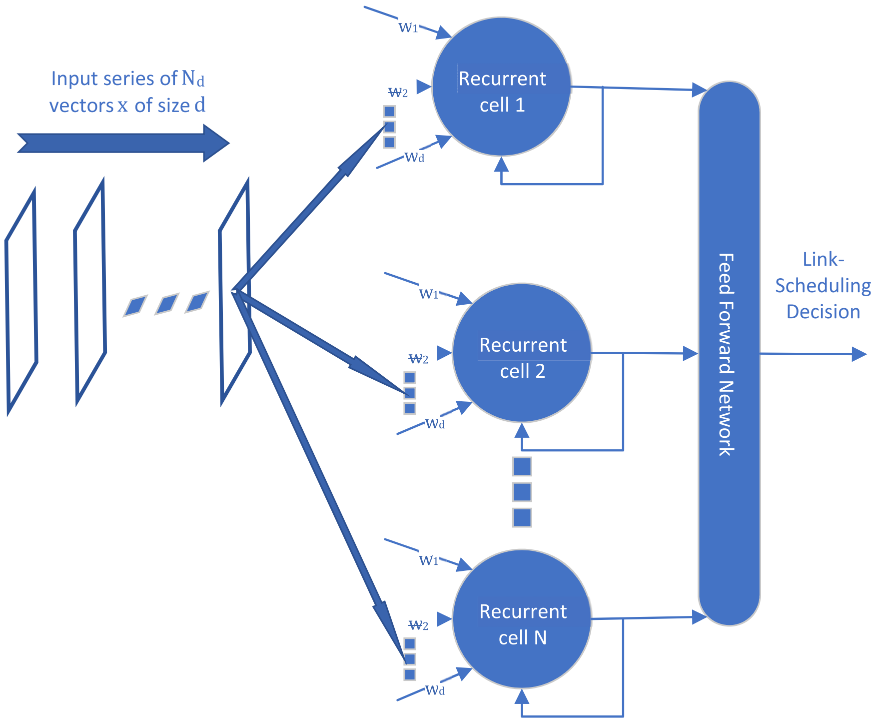

Fig. 4. The proposed SSM for D2D link-scheduling.

describing.

a pair and previous output state $\mathbf{y_{j-1}}$ as follows

$$ij = \sigma\ (Wixj + Hiyj - 1) \tag{2}$$

$$\mathbf{oj} = \sigma\ (\mathbf{Woxj} + \mathbf{Hoyj} - \mathbf{1}) \tag{3}$$

$$\mathbf{a_j} = \tanh\left(\mathbf{W_a x_j} + \mathbf{H_a}\ \left(\mathbf{i_j} \odot \mathbf{y_{j-1}}\right)\right) \tag{4}$$

$$\mathbf{y_j} = \left(1 - \mathbf{o}_j\right) \odot \mathbf{y_{j-1}} + \mathbf{o_j} \odot \mathbf{a_j} \tag{5}$$

where $\mathbf{W}_*$ and $\mathbf{H}_*$ are the trainable input and hidden states' weights, respectively, for each gating signal indicated by $*$, σ is the sigmoid function and $\odot$ is the element-wise multiplication operator. The input gate $\mathbf{i_j}$ decides whether the previous hidden state affects the new one or not using the current input and previous hidden state. The output gate $\mathbf{o_t}$ provides a weighting between the internal state $\mathbf{a_t}$ and the previous output state $\mathbf{y_{j-1}}$ to produce the new output state $\mathbf{y_j}$. Fig. 5 illustrates how the gates and inputs interact.

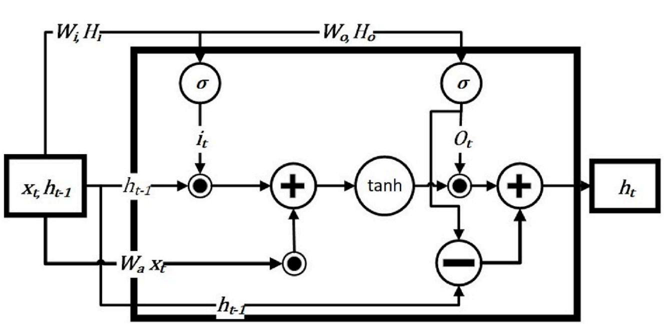

Fig. 5. Structure of the GRU cell.

4. Simulation results

4.1. Simulation setup

The simulated network layouts are generated using the code from (Ba and Caruana, 2014) and the parameters described in (Prabhavalkar et al., 2017) for proper comparison. The simulator uses the ITU-1411 outdoor channel model where distance-dependent pathloss and shadowing are considered (Recommendation, 1411-11). The range of the channel gains produced by this model will depend on the dimensionality of the simulation environment, which is chosen to be a 500 × 500 square area, the separation distance between the Tx and Rx members of a D2D pair, which is chosen to be uniformly distributed between 2 and 65 m, as well as the lognormal shadowing standard deviation σ_s. The default number of D2D pairs is chosen to be 50 in the square coverage area and the number of generated training samples is 500 while 1000 test samples are used. The three nearest neighbors ($K = 3$) were chosen to construct the descriptors since the performance of the classifier did not improve significantly with the increase of the number of neighbors beyond $K = 3$. The speed of D2D pairs is kept constant at 5 km/h, which is a typical

Table 1
Summary of simulation and model parameters.

Network Layout Parameters	
Square area side length	500 m
D2D distance	2 – 65 m
Noise spectral density	−169 dB m/Hz
Bandwidth	5 MHz
Carrier frequency	2.4 GHz
Antenna height	1.5 m
Active link transmit power	40 dB m
SSM Parameters	
Hidden GRU Units	30
Output neurons	2
Epochs	30
Batch size	64
Optimizer	Adam

pedestrian speed. All the other parameters are summarized in Table 1.

The performance is quantified as a percentage of the sum rate obtained from the optimal scheduler FPLinQ (Shen and Yu, 2017). The reported percentages are the averages of the 1000 test samples. It is impotent to note that different scheduling combinations can end up with the same sum rate. Fig. 6 shows the distribution of performances for 1000 test samples when there are 100 D2D pairs in each. It can be readily seen that in some cases the SSM even outperforms FPLinQ. This can happen because the SSM does not exactly replicate the FPLinQ output, nor has all its given inputs, which include the channel quality indicators. Hence, there is a chance that relying only on the path-loss information can produce better sum rate if the channel estimation was misleading. It is important to also note that FPLinQ does not provide a global optimal solution, but rather, a very good approximation to it so it is possible that other techniques may outperform its performance in specific runs. Having said that, in this work, it was always found that the average performance of the proposed SSM technique is worse than FPLinQ in all the conducted experiments as will be shown in the sequel. Since the complexity and the number of required training samples required by the Kernel model are prohibiting, we will establish comparisons only with the graph embedding approach. This comparison will be quantitatively evaluated using the Wilcoxon Signed-Rank Test (Friedman and Hastie, 2009). The p-value result from the test describes how likely the hypothesis that both sets of experiments come from different distributions, i.e., the smaller the p-value, the more similar the results of the two sets of experiments are. There are two commonly used thresholds for the p-value to decide on the similarity of the experiment sets, which are 0.05 and 0.1. So, we will consider values more than 0.1 to be similar and less than 0.05 to be different, and in-between to be almost similar. The fast decision-making process of the proposed model allows the assumption of a quasi-static network layout.

This assumption allows the network to be trained with a multitude of scenarios but without assuming correlation between the different layouts. Hence, the locations can be assumed completely random from one instance to another. On the other hand, when mobility is taken into consideration as an input piece of information then the correlation in time needs to be explored for improving the decision-making process.

4.2. Simulation results for quasi-static D2D networks

4.2.1. Comparison of generalizability

Generalizability refers to the ability of the model to perform well in testing environments other than those seen during the training phase. Table 2 shows the percentage of the sum rate obtained from the optimal scheduler FPLinQ for the proposed model compared to that of graph embedding as reported in (Prabhavalkar et al., 2017) assuming different number of D2D pairs. Note that the model is trained using 50 D2D pairs as described earlier and tested using various numbers of D2D pairs. Clearly, the proposed model is resilient to changes in the number of D2D pairs in the test compared to those in the training set.

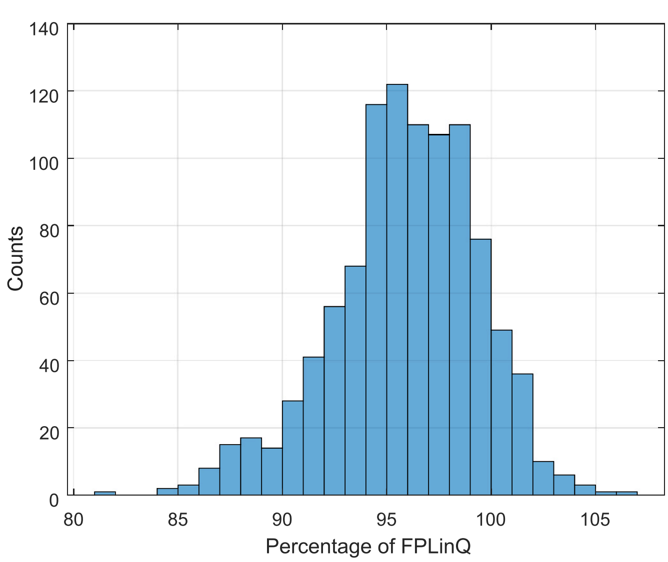

Fig. 6. Distribution of sum rate performance for SSM for 1000 test scenarios of 100 D2D pairs.

This is due to the fact that the feature used is independent of the number of D2D pairs and relies only on the three ($K = 3$) nearest neighbors. In fact, as the number of D2D pairs increases, the decisions based on distances between the neighbors become more effective in the resource scheduling decisions and hence, the notable improvement in performance when the number of D2D pairs increases. The graph embedding, on the other hand, does not base the internal NN calculations on raw ordered distances, but there is the embedding phase, which non-linearly transforms the distances in the graph into a quantized vector representation. This, in turn, compromises the information retained as the number of pair-wise distances increases.

The Wilcoxon test for this comparison results in a p-value of 0.75, which indicates that the proposed SSM performs as well as graph embedding but with lesser required resources and parameters as will be shown in the sequel. The reason for this overall similarity when comparing the experiments from both models is that the SSM performs better when the number of D2D pairs increases, and worse when it is low. Being able to perform well in different scenarios will decrease the requirement to re-train the model and will maintain acceptable performance for different network scales decreases the requirement of having to acquire labels for training. In other words, running the optimal classifier during the training phase to obtain labels for training will not be required frequently since the proposed model scales well without compromising the performance.

Hence, the impact of requiring the optimal classifier's decisions during the training phase will have minimum impact.

Another comparison for generalizability is carried out for different shadowing standard deviations and the results are shown in Table 3. Since the learning is done only using distance information, the model will be challenged to generalize for scenarios when an unknown component distorts the communication. Table 3 shows lesser sum rates as the shadowing standard deviation increases. The Wilcoxon test p-value reports a value of 0.0625, which suggests that the proposed model is almost similar to graph embedding. The comparison is clearly in favor of the proposed model since it has lesser complexity and has fewer parameters as will be shown below.

4.2.2. Comparison of scalability

Scalability refers to the ability of the model to perform well under different scenarios given that there is training data simulating the actual testing or working network environment but without discrimination between the training and testing environments. In this case, it is beneficial to observe the scalability of the SSM learning classification accuracy with respect to the indirect objective of maximizing the sum rate. Table 4 shows how the proposed model performs for various numbers of D2D pairs from both the classification accuracy and the achievable sum rate perspectives in comparison with the graph embedding network. Clearly, the proposed SSM gives almost similar overall performance to graph embedding with a p-value of 0.0625 for the.

Wilcoxon test with better performance in dense networks. The

Table 2
Generalizability of the proposed model and graph embedding for different numbers of D2D pairs.

Number of D2D Pairs	10	30	50	80	100
Graph Embedding	0.96	0.97	0.94	0.9	0.85
Proposed Model	0.94	0.93	0.92	0.96	0.96

Table 3
Generalizability of the proposed model and graph embedding for different shadowing standard deviation values.

Shadowing σ_s (dB)	0	3	5	8	10
Graph Embedding	0.94	0.93	0.87	0.81	0.75
Proposed Model	0.93	0.9	0.84	0.74	0.67

Table 4
Scalability of the proposed model for different number of D2D pairs vs. graph embedding.

Number of D2D Pairs		10	30	50	80	100	500
Classifier Accuracy	Graph Embedding	0.88	0.81	0.82	0.82	0.83	0.89
	Proposed Model	0.85	0.85	0.87	0.89	0.9	0.95
Average Sum Rate	Graph Embedding	0.98	0.96	0.95	0.93	0.92	0.87
	Proposed Model	0.94	0.94	0.95	0.94	0.96	0.94

performance of the SSM is better in the case of higher numbers of D2D pairs because as the number of D2D pairs increases in a fixed area the distances tend to become more representative of the channel loss. In other words, the scheduling decisions become more accurate for shorter separations between the D2D pairs. The reason that SSM performs better in denser networks is that it uses raw distance data without any mapping.

Fig. 7 shows the convergence speed of the SSM training phase for both the average sum rate and classifier accuracy. The figure shows that convergence starts after approximately 15 epochs, which is half the time reported by the graph embedding method. Also, there is no mismatch between the sum rate performance and the training accuracy. This also shows consistency between the model's performance in both classification and scheduling, which proves to be very helpful in practical scenarios. On the other hand, in (Prabhavalkar et al., 2017), the graph embedding method shows an inconsistent behavior, which makes the anticipation of an improvement in the scheduling performance from an improvement in the model performance unclear.

4.2.3. Comparison of model complexity and computational requirements

We conclude this subsection by comparing the complexity of the proposed approach against both graph and kernel embedding. Complexity directly impacts the model's computational requirements and, in turn, affects the speed with which the link-scheduling decisions come through. Table 5 shows such comparison. From the table, it is evident that the proposed model achieves the same scheduling accuracy as that reported by (Prabhavalkar et al., 2017) while requiring only 29.5% of its parameters, 31.6% of its multiplication operations and is not confined to a specific number of D2D pairs to schedule. Similarly, the proposed model shows a negligible 4.4% loss in sum rate while gaining a significant decrease in the number of parameters, which is only 0.85% of those required by the kernel model and a similar significant decrease in the number of multiplication operations to only 11% of the latter as well. As introduced before, the reduction in complexity is a result of the SSM built-in flexibility in modelling dependent input data, and using raw yet informative distance information, which minimizes the learning burden on the network to model the network layout. Finally, when compared to FPLinQ which has a complexity in the order of O(N^2), the complexity in SSM is in the order of O(N).

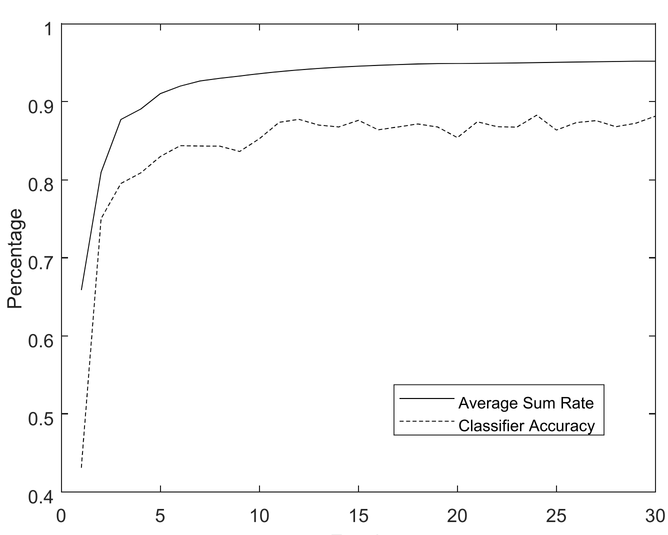

Fig. 7. Training convergence speed with 50 D2D pairs and 500 training sample layouts.

4.3. Simulation results for dynamic D2D networks

In this section, we describe the inclusion of the spatio-temporal dependencies. As mentioned in Section 4.1, the link scheduling problem needs to deal with dynamicity in both space and time due to mobility of the D2D pairs. Fig. 1 shows how the space information is represented by distances calculated around each D2D pair as detailed in Section 3.1 and Fig. 2 illustrates how the spatial locations of the D2D pair change throughout time. The inputs are processed starting with the input at t_0 and up to t_{Ns}. The decision takes place for each D2D pair for every time step throughout the presence of the D2D pair within the coverage area.

Simulation results are generated using a modified version of the code from (Ba and Caruana, 2014). The code was modified to generate one initial random layout of the network, and then mobilize each Tx-Rx pair in a random direction with a certain speed. All pairs are forced to stay within the network coverage region by bouncing off the boundaries if they reach it. The network parameters used are those described in (Prabhavalkar et al., 2017) to allow for proper comparison of performance. All simulation parameters are the same as those mentioned in Section 4.1 with the speed set to a constant pedestrian speed of 5 km/h.

4.3.1. Comparison of generalizability

Table 6 shows the generalizability performance measure for SSM in the dynamic network scenario. The results of the proposed SSM for

Table 5
Comparison between the number of parameters and multiplication operations for different NN structures.

Point of Comparison	Kernel Model (Ba and Caruana, 2014)	Graph Embedding (Prabhavalkar et al., 2017)	Proposed SSM
Main DL Technique	Kernel method and DNN	Graph embedding and DNN	Recurrent NN
Required Training Samples	Hundreds of thousands	Hundreds	Hundreds
Training Method	Unsupervised	Supervised or Unsupervised	Supervised
Number of layers	For each D2D pair: * 2 Convolutional Filters 63 × 63 * 2 Hidden Layers with 30 neurons * 1 output neuron	* 1 Embedding layer to 32 * 1 Hidden Layer with 64 neurons * 2 × 50 Output neurons	* 16 input descriptors * 30 GRU cells * 2 output neurons
Approximate number of learnable parameters	0.8 M	9.5 k	2.8 k
Approximate number of multiplication operations 90 k		19 k	6 k

Table 6
Generalizability of the quasi-static and dynamic models for different numbers of D2D pairs.

Number of D2D Pairs	10	30	50	80	100
Quasi-static Model	0.96	0.97	0.94	0.9	0.85
Dynamic Model	0.93	0.90	0.95	0.93	0.90

quasi-static modelling using only spatial correlations are compared to that of the proposed SSM exploiting both spatio-temporal ones. Note that the model is trained using the default 50 D2D pairs as mentioned earlier. Since the proposed model tracks the mobility of each D2D pair throughout time assuming dependency between the different time instants, the model is left with very limited data to learn from when the number of D2D pairs is small. On the other hand, the benefit of adding more data is clear and does not require adding any extra learners to the SSM method to achieve better performance with significantly low computational requirements as already shown before. Another comparison for generalizability is carried out for different shadowing standard deviations and the results are shown in Table 7. Of course, as the shadowing standard deviation level increases, the assumption of correlated time samples becomes less effective and there is a notable deterioration in performance.

4.3.2. Comparison of scalability

Table 8 compares the results of both types of proposed models. The scalability seems to improve more in dense networks in the dynamic case than in the quasi-static case. This results from the fact that time correlations are more impactful than spatial correlations alone. Also, the correlations between the consecutive decisions in time are more stable when the network is dense than in the less dense networks where the distance information are not sufficient to make decisions because of the wide spacing between the different D2D pairs.

5. Unsupervised D2D link scheduling

The proposed SSM showed good generalizability, but it still requires some labelled training data. In scenarios where a new area is being covered, or when the coverage is done by a remote station with limited capabilities then unsupervised learning becomes a necessity. From the previous experiments, it was noted that in denser networks, the decision-making relies more on the separating distance between the D2D pairs. As a further investigation, a density-based estimation of the distribution of separating distances between the pairs for different scenarios is explored. To achieve this, an empirical histogram is explored and then a performance evaluation is done using a simple threshold approach. Fig. 8 shows the distribution of separating distances for a low-density network of 10 D2D pairs and a high-density network of a 100. The skewness of the distributions becomes clearer in the high-density case due to the confinement in the coverage area and increased interference. Hence, the optimal decisions tend to favor the stronger links.

With the simple proposed thresholding approach, distributed decision-making can be enabled. The proposed approach enables each D2D pair to decide whether they can initiate a communication link at a given moment or not without relying on a centralized decision maker. All what is required from the central node is to broadcast the threshold value and each pair can decide based on their separating distance.

Clearly, for an unsupervised approach, the distributions of each decision will be unknown. The only available observation is the uniform

Table 7
Generalizability of the proposed model for a dynamic D2D network with shadowing.

Shadowing σ_s (dB)	0	3	5	8	10
Quasi-static Model	0.94	0.93	0.87	0.81	0.75
Dynamic Model	0.95	0.91	0.84	0.72	0.63

Table 8
Scalability of the proposed model for quasi-static and dynamic D2D network for different number of D2D pairs.

Number of D2D Pairs	10	30	50	80	100
Quasi-static Model	0.94	0.94	0.95	0.94	0.96
Dynamic Model	0.94	0.95	0.95	0.96	0.98

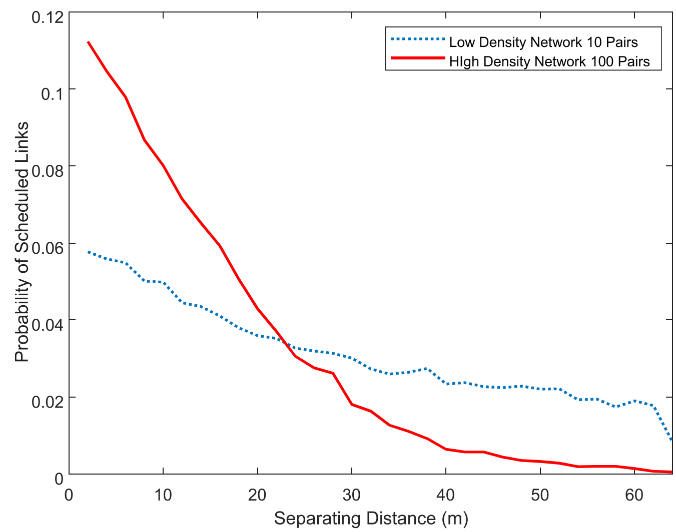

Fig. 8. Probability of scheduled links versus the separating distances in case of a low-density network with 10 pairs and a high-density network with 100 pairs.

distribution for the separating distances. The safest decision would hence be the midpoint of the range. The performance relative to the optimal classifier is shown in Table 9. It is clear that the performance is much better in the high-density networks. This fact can be very useful for distributed processing since all what the core network needs to share with the users is the decision threshold as mentioned earlier. Although this approach requires almost no computations at all, it comes at the cost of performance. If the performance is crucial for a given system, then the supervised approach is recommended.

It is worth mentioning, however, that the extra complexity is encountered during the offline training phase, while the online performance is unaffected by the system training requirements. The offline training and model update can be done during the non-busy hours of the day if required. This requirement comes only whenever there is a significant change in the layout of traffic of the covered area. Otherwise, there will not be any need for such an update.

6. Conclusion

This paper tackled the link scheduling problem in D2D networks. The proposed SSM solution uses a DL-based approach to tackle the problem using only raw distance information. Also, the paper showed the benefit of taking into account the correlation between time samples when a space-time model is considered where the information required is only distance information. The proposed model achieved approximately 90% reduction in the number of required parameters and multiplication operations required by the previously proposed simplest model in the literature while achieving statistically better generalizability

Table 9
Unsupervised vs supervised link scheduling using only the separating distance.

Number of D2D Pairs	10	30	50	80	100
Quasi-static Model	0.96	0.97	0.94	0.9	0.85
Dynamic Model	0.93	0.90	0.95	0.93	0.90
Unsupervised Link Scheduling	0.70	0.77	0.80	0.85	0.88

performance. Also, in other tests, it achieved a statistically comparable performance, which is on average equivalent to 92% of the sum rate performance of the optimal scheduler that uses CSI and solves a complex MILP optimization problem to perform resource allocation. Finally, analyzing the decision-making of the proposed SSM solution resulted in formulating a very simple threshold decision-making that works in an unsupervised manner and achieves acceptable results. This approach requires very limited computational resources and can be suitable for remote base stations with limited capabilities. The promising outcomes of the proposed scheduling process as well as the significant reduction in computational requirements will translate to a reduction in the power requirements of future base stations. Hence, D2D scheduling can be achieved using remote stations that can run on sustainable and renewable energy sources, thus opening the door for a wider adoption of green communication. This will also lead to a faster deployment of applications that can benefit from D2D scheduling such as vehicular communication or cached media streaming.

Credit author statement

Ahmed Elsheikh: Conceptualization, Methodology, Software, Writing – original draft. Ahmed S. Ibrahim: Conceptualization, Writing – review & editing, Validation. Mahmoud H. Ismail: Writing – review & editing, Validation, Supervision

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The work of Ahmed Elsheikh and Mahmoud H. Ismail is supported by the Smart Cities Research Institute (SCRI) at the American University of Sharjah under grant EN0281:SCRI18-07. The work of Ahmed S. Ibrahim is supported in part by the National Science Foundation under Award No. CNS-2144297.

References

Ba, J., Caruana, R., 2014. Do deep nets really need to be deep? Adv. Neural Inf. Process. Syst. 27, 2654–2662.
Chen, M., Challita, U., Saad, W., Yin, C., Debbah, M., 2019. Artificial neural networks-based machine learning for wireless networks: a tutorial. IEEE Communications Surveys & Tutorials 21, 3039–3071.
Cui, W., Yu, W., 2020. A clustering approach to wireless scheduling. In: IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). IEEE, pp. 1–5.
Cui, W., Shen, K., Yu, W., 2019. Spatial deep learning for wireless scheduling. IEEE J. Sel. Area. Commun. 37, 1248–1261. https://doi.org/10.1109/JSAC.2019.2904352. arXiv:1808.01486.
De Boer, P.-T., Kroese, D.P., Mannor, S., Rubinstein, R.Y., 2005. A tutorial on the cross-entropy method. Ann. Oper. Res. 134, 19–67.
Friedman, J., Hastie, T., Tibshirani, R., 2009. The elements of statistical learning. In: Springer Series in Statistics. Springer, New York.
Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016. Deep Learning, 1. MIT press Cambridge.
Haykin, S., 2010. Neural Networks and Learning Machines, third ed. Pearson Education.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735. URL:
Jourdan, D., Deyst, J., Win, M., Roy, N., 2005. Monte Carlo Localization in Dense Multipath Environments Using UWB Ranging. IEEE International Conference on Ultra-Wideband, pp. 314–319. https://doi.org/10.1109/ICU.2005.1570005.
Jozefowicz, R., Zaremba, W., Sutskever, I., 2015. An empirical exploration of recurrent network architectures. In: International Conference on Machine Learning. PMLR, pp. 2342–2350.
Lee, M., Yu, G., Li, G.Y., 2020. Graph embedding-based wireless link scheduling with few training samples. IEEE Trans. Wireless Commun. 20, 2282–2294.
Liu, Q., Liu, R., Wang, Z., Han, L., Thompson, J.S., 2021. A V2X-integrated positioning methodology in ultradense networks. IEEE Internet Things J. 8, 17014–17028.
Luo, F.-L., 2020. Machine Learning for Future Wireless Communications. John Wiley & Sons.
Prabhavalkar, R., Rao, K., Sainath, T.N., Li, B., Johnson, L., Jaitly, N., 2017. A Comparison of Sequence-To-Sequence Models for Speech Recognition. Interspeech, pp. 939–943.
Recommendation, I.T.U.-R.P., 2009. Propagation Data and Prediction Methods for the Planning of Short-Range Outdoor Radiocommunication Systems and Radio Local Area Networks in the Frequency Range 300 MHz to 100 GHz, Standard. International Telecommunications Union, Geneva, CH, 1411-11.
Salarian, M., 2015. Accurate localization in dense urban area using Google street view images. SAI Intelligent Systems Conference (IntelliSys) 485–490.
Shen, K., 2020. Fractional Programming for Communication System Design. Ph.D. thesis, University of Toronto (Canada).
Shen, K., Yu, W., 2017. FPLinQ: a cooperative spectrum sharing strategy for device-to-device communications. In: IEEE International Symposium on Information Theory (ISIT). IEEE, pp. 2323–2327.
Shen, Y., Shi, Y., Zhang, J., Letaief, K.B., 2020. Graph neural networks for scalable radio resource management: architecture design and theoretical analysis. IEEE J. Sel. Area. Commun. 39, 101–115.
Soldani, D., Pentikousis, K., Tafazolli, R., Franceschini, D., 2014. 5G networks: end-to-end architecture and infrastructure [guest editorial]. IEEE Commun. Mag. 52, 62–64.
Ye, H., Li, G.Y., Juang, B.-H.F., 2019. Deep reinforcement learning based resource allocation for V2V communications. IEEE Trans. Veh. Technol. 68, 3163–3173.
Yu, S., Wang, C., Yu, Z., Li, X., Cheng, M., Zang, Y., 2021. Deep regression for LiDAR-based localization in dense urban areas. ISPRS J. Photogrammetry Remote Sens. 172, 240–252.

Dr. Ahmed Elsheikh is currently an assistant professor at the Mathematics and Engineering Physics Department at Cairo University, Giza, Egypt. He received the B·Sc. degree in electronics and electrical communications engineering from Cairo University, Giza, Egypt, in 2011. His MSc. From the Mathematics and Engineering Physics, Cairo University, Giza, Egypt in 2014. He received the Ph.D. degree in Mathematics and Industrial engineering from the University of Montreal, Montreal, Quebec, Canada in 2018. Dr. Ahmed's research interests are focused on applied machine learning in various domains such as Engineering Physics, Industrial Engineering, and Communications.

Dr. Ahmed S. Ibrahim is currently an associate professor at the Electrical and Computer Engineering Department at Florida International University (FIU), Miami, FL, USA. He received the B.S. and M.S. degrees in electronics and electrical communications engineering from Cairo University, Cairo, Egypt, in 2002 and 2004, respectively. He received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA, in 2009. He is a recipient of the NSF CAREER Award in 2022. Prior to joining FIU, Dr. Ibrahim was an assistant professor at Cairo University, wireless research scientist at Intel Corporation, and senior engineer at Interdigital Communications Inc. Dr. Ibrahim's research interests span various topics of wireless systems including drone-assisted millimeter wave communications, vehicular communications, and rethinking wireless networks through the lens of Riemannian geometry.

Mahmoud H. Ismail (S′00-M′07-SM'15) received the B·Sc. degree (with highest honors) in Electronics and Electrical Communications Engineering, the M.Sc. degree in Communications Engineering both from Cairo University, Egypt, in 2000 and 2002, respectively, and the Ph.D. degree in Electrical Engineering from The University of Mississippi, MS, USA, in 2006. From August 2000 to August 2002, he was a Research and Teaching Assistant in the Department of Electronics and Electrical Communications Engineering at Cairo University. From 2004 to 2006, he was a Research Assistant in the Center for Wireless Communications (CWC) at the University of Mississippi. He is currently a Full Professor (on leave) at the Department of Electronics and Electrical Communications Engineering, Cairo University and an Associate Professor at the American University of Sharjah, Sharjah, UAE. He was also a Systems Engineering Consultant at Newport Media Inc. (now part of Microchip) Egypt Design Center in Cairo from 2006 to 2014. His research is in the general area of wireless communications with emphasis on performance evaluation of next-generation wireless systems and communications over fading channels. He is the recipient of the University of Mississippi Summer Assistantship Award in 2004 and 2005, The University of Mississippi Dissertation Fellowship Award in 2006, The University of Mississippi Graduate Achievement Award in Electrical Engineering in 2006 and the Best Paper Award presented at the 10th IEEE Symposium on Computers and Communications (ISCC, 2005), La Manga del Mar Menor, Spain. He served as a reviewer for several refereed journals and conferences and he is a Member of Sigma Xi and Phi Kappa Phi.