Machine Learning for Power Allocation of a D2D Network

Junxiu Zhou
Department of Computer Science
Northern Kentucky University
Highland Heights, KY, USA

Xian Liu
Department of Systems Engineering
University of Arkansas at Little Rock
Little Rock, AR, USA

Changcheng Huang
Department of Systems and Computer Engineering
Carleton University
Ottawa, ON, Canada

***Abstract*—Resource optimization for small-cell wireless networks is more complicated than the traditional applications. The solution needs to be delivered promptly to respond the highly dynamic temporal and spatial variations. It seems that the machine learning strategy is more flexible and adaptive than the conventional optimization methods, since ML has the potential to find the implicit function relationship between arbitrary input data and output results. In this work, we focus on a generic D2D network and to show the effectiveness of ML apply to solve the power optimization problem with different optimization models. The research spans over all stages such as analysis, design, implementation, and validation. It is shown that the ML method has achieved several benchmarks in terms of QoS metrics for different optimization models.**

Keywords—Deep learning, machine learning, neural networks, optimization, wireless communications.

I. INTRODUCTION

The *machine learning* (ML) approach, as a fundamental methodology in modern *artificial intelligence* (AI) discipline, has gained strong interests in various areas, such as wireless networks [1]. In many situations, ML can provide a workable solving strategy and produce a set of useful results in a reasonably quick manner. This is particularly important in a *device-to-device* (D2D) communication task, where the traffic is highly dynamic, and the topology keeps changing [2].

D2D allows direct communication between mobile users, providing proximity-based services in wireless networks [3]. It has the advantages in increasing area spectral efficiency, improving cellular coverage and lower latency, and reducing power consumption. However, in D2D communication networks, there are additional impairments caused by path-loss, shadowing, and multipath scattering. If the wireless device is also mobile, then the Doppler effect must be taken into account too. How to design a workable scheme to solve the networking related issues in D2D networks in a timely manner has been a challenge in practice. Although many sophisticated approaches have been proposed to deal with the resource allocation issues of D2D wireless networks, and analytically these approaches are established on solid mathematical structures, their practical applications are usually restricted in field [4]. The reasons are multiple, but primarily the limited applicability is due to the limited CPU capacity and/or storage capacity in small devices. It is highly desirable to get rid of those computation-intensive schemes and develop some economical and quick schemes to solve the same problem. In this paper, we try to customize the ML approach to solve an optimization problem commonly arisen in small-cell wireless networks. Guided by the principle of ML, through extensive practices in analysis, design, implementation, and validation, we have gained some important insights in the investigated model.

The rest of this paper is organized as follows. In Section II, the related works to this study is overviewed. In Section III, a new model for resource optimization is presented. Then, in Section IV, the essentials of ML are introduced. Next, in Section V, the detailed practices on ML are described and extensive experimental results are presented with detailed discussions. Finally, the conclusion is put in Section VI.

II. RELATED WORK

The primary goal of power allocation in the D2D networks is to ensure the communication signals to have the sufficient coverage by limiting the interference caused by concurrently active D2D users, while optimizing some performance merits in the system level, such as the sum rate over all links [3]. There have been many approaches to solve the power allocation problem. Among them, recently, the ML-based methods have received great attention, because of their feasibility of offline training and the real-time performance of online testing. Several inspiring results has been reported. For example, Lee *et al.*[5] proposed to feed the channel matrix into a *convolutional neural network* (CNN) to optimize the power allocation task. Xu *et al.* [6] proposed to make use of deep reinforcement learning on solving power-efficient power allocation problem in *cloud radio access networks* (C-RANs). The paper [7] is the first to adopt a *deep neural network* (DNN) based on *universal approximation theorem* (UAT) [8] to optimize the power allocation problem. They used a DNN model to approximate the performance of the iterative-based WMMSE method [9] for power allocation. They validated the effectiveness of the DNN-based approach for approximating the iterative-based method and the computational time advantage of DNN-based method. Since then, several papers have been proposed to use DNN-based approaches to allocate power resources in wireless networks. Inspired by [4, 7], Zappone *et al.* [10] proposed to use DNN to optimize the *global energy efficiency* (GEE) maximization power allocation model in an interference-limited network. Their DNN is trained by the results of a *sequential fraction programming* (SFP) optimization method instead of WMMSE. Liang *et al.* [11] proposed to use a DNN to optimize the power resource by directly maximize the sum-rate as the loss function. They also took noise power and channel coefficients in addition to channel gain as input to their DNN, in order to train the DNN to have the ability to handle a range of noise power levels. In addition, they ensemble multiple DNNs together to build an ensemble network to achieve better output power values. Eisen *et al.* [12] modeled the power resource allocation problems, such as the simple channel model and the interference channel model, as one generic formulation. They then combine the conventional indirect optimization method with DNN to optimize the generic formulation. That is, to solve the non-convex generation formulation, they employed the Lagrangian duality method and a DNN model. Specifically, the backpropagation process of DNN is taken place by the Lagrangian duality optimization process of the generic formulation. Therefore, DNN needs no training examples during the training process.

We note that, however, all the above-mentioned works did not take the *quality of service* (QoS) constraint into account in the optimizing models. In D2D networks, optimizing the power alone may reduce the index of QoS.

The QoS-based power allocation problem is extremely important for the next-generation wireless networks, since they are expected to support very high data rates and radically new applications. With the additional QoS constraint considered, the above methods may not work well. In this work, we consider the additional QoS constraint in the learning process to optimize the power allocation problem. It should be mentioned that the work in [13] also used DNN to optimize the QoS-aware power management problem. However, the present work proposes a new model to seek the similar merit to the *weighted sum-rate* (WSR) function, while holding the convexity computationally.

III. System Model and Problem Formulation

In this section, we first describe the system model for a generic network subject to interference, and then propose the optimization objective formula for our power allocation problem.

A. System Model

We consider a wireless network deployed in a region represented by a disc with radius R_c. This generic network consists of N pairs of *transmitters* (Tx) and *receivers* (Rx). In the present work, we consider a network consisting of the same number of transmitters and receivers, which is one of the most representative configurations of the D2D communication paradigms. We use $\mathcal{C}_T = \{1,2,\ldots,N\}$and $\mathcal{C}_R = \{1,2,\ldots,N\}$ to denote the index sets of Tx and Rx, respectively.

The topology of the wireless network is shown in Fig. 1, where the solid lines represent the desired transmission links, while the dotted lines represent the interfering links. Note that the pattern in Fig. 1 is just the topology, rather than the actual layout. In practice, the layout of a D2D network is usually random, although there are some location restrictions to avoid the singularity and other anomalies.

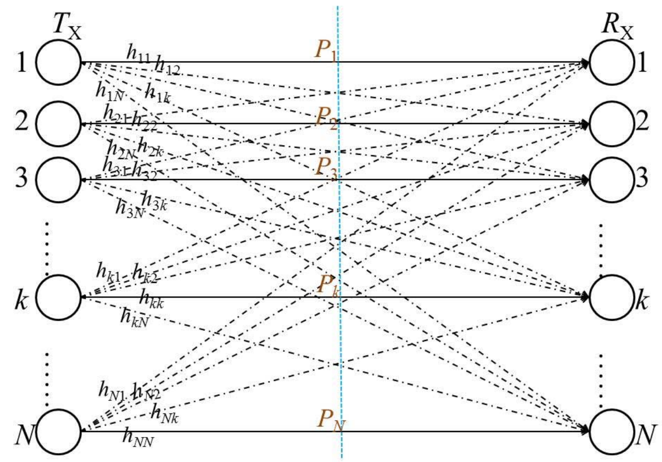

Fig. 1. Generic wireless network structure.

TABLE I. The List of Main Notations

Notation	Description
h_{jk}	Channel gain of the interference link jk
h_{kk}	Channel gain of the desirable link paired with Tx k and Rx k
N	Number of Tx-Rx pairs
r_{jk}	The length of link jk (meter)
u_k	SINR of Rx k (dB)
$u_{k,\min}$	QoS threshold of SINR (dB)
P_k	Transmitter power of Tx k (dBm)
α_{jk}	Path-loss exponent of link jk
σ_k^2	Noise power of Rx k

The main notations are listed in Table I. Other notations will be defined in the relevant context. In Table I, the term "link ij" means the link from Tx i to Rx j. The term "channel gain" refers to the small-scale fading. The Rayleigh fading is adopted in the analysis throughout this paper. Thus the channel gains h_{kj} and h_{kk} follow the exponential distribution. The conventions commonly used in the literature of wireless communications are also adopted, such as unit mean and the i.i.d. condition. This way, the large-scale fading due to the path-loss will be described with a power-law term. To concentrate on the key concept, the shadowing effect is included in the large-scale fading with appropriately adjusted parameters.

In the present model, half-duplex is assumed, i.e., a node cannot receive signals while simultaneously transmit signals. For example, at a particular moment, the link from Tx 3 to Rx 7 is different than the link from Tx 7 to Rx 3. This implies that, in general, it is not necessarily to have $b_{jk} = b_{kj}$, $b \in \{h, r,$

$\alpha\}$. Also, $r_{kk} \neq 0$, since it represents the distance from *T*x *k* to *R*x *k*. σ_k^2 characterizes the *additive white Gaussian noise* (AWGN). With these elaborations, the *signal-to-interference-plus-noise ratio* (SINR) for the receiver *k* is expressed as:

$$u_k \triangleq \frac{h_{kk}\left(P_k/r_{kk}^{\alpha_{kk}}\right)}{\sigma_k^2+\sum_{j=1,j\neq k}^{N} h_{jk}\left(P_j/r_{jk}^{\alpha_{jk}}\right)} \quad (1)$$

where usually $1.6 < \alpha_{jk} < 9$. Note that in (1) h_{jk}, h_{kk}, r_{jk}, and r_{kk} are *random variables* (RVs), while the variables P_j and P_k are the entities to be optimized (referred to as the *decision variables* in optimization literature).

B. Problem Formulation

In the present work, we develop a QoS-aware scheme to optimally allocate the power for all transmitters. There are several ways to characterize the merit of scheme. One of the most popular ones is the *weighted sum-rate* (WSR). The original WSR model did not explicitly include the QoS constraints. The WSR model augmented by the QoS constraints can be expressed as follows:

(Model-1)

$$maximize\ \sum_{k=1}^{N} w_k \log_2(1+u_k) \quad (2)$$

$$s.t. \quad u_{k,min} \leq u_k \quad (3a)$$

$$0 \leq P_k \leq P_{k,max} \quad (3b)$$

$$k = 1,2,\dots,N$$

where u_k is defined in (1). w_k denotes the bandwidth. $P_{k,max}$ refers to the allowed maximum power of *k*.

The WSR provides a clear insight since is directly related to the Shannon's capacity. However, the optimization model with WSR as the objective function is non-convex. The non-convexity is unfavorable for many reasons of either analytical or computational. Analytically, the convexity guarantees the global optimality rather than local. Computationally, the convex problem is polynomially solvable. It is highly desirable to seek a similar merit as the objective function while holding the convexity. In the present work, we consider the following QoS-aware power optimization model:

(Model-2)

$$\min_{P_k,k=1,\dots,N} \sum_{k=1}^{N} w_k[log_2(1+\mu_k)]^{-1} \quad (4)$$

$$s.t. \quad u_{k,min} \leq u_k \quad (3a)$$

$$0 \leq P_k \leq P_{k,max} \quad (3b)$$

$$k = 1,2,\dots,N$$

The proof of convexity is omitted here due to the space limit. A further observation is, regarding the intermediate variable SINR u_k, the objective function is monotonic. Due to the overall convexity, the constraint eq. (3a) is redundant, since any numerical optimization algorithm is also monotonic, either "up-hill" or "down-hill". Therefore, only the power constraints (3b) are needed. This is really desirable, since (3b) represents a box-shape feasible region for the original decision variables, viz. the transmit power. Dropping the constraint (3a), Model-2 will be referred to as Model-3 in the sequel.

IV. Overview of Neural Network Model

In general, the power allocation problem can be solved by the numerical optimization routines included in most software toolkits (e.g., Matlab). Most such routines comprise a collection of algorithms, and each algorithm consists of multiple iterations. In principle, the constrained optimization problems can be solved in one of two basic approaches: the direct method or the indirect method. In the direct method, the iterations are carried out with in the feasible region, which is typically a sub-space with high dimensions. When the searching point hits the boundary, the searching direction is adjusted by an angle. In the indirect method, the constraint function is included into the objective function to form an augmented objective function. Then an unconstrained optimizer is employed. However, no matter whether using the direct or indirect methods, usually a large number of iterations are needed to solve the power allocation problem. The computational time is often excessive for the large-scale problems. The traffic flows in practical wireless networks, however, are highly dynamic and most likely random. There is a real-time requirement for the problem solver. The *machine learning* (ML) discipline just provides a new methodology along the avenue of lunching quicker solvers.

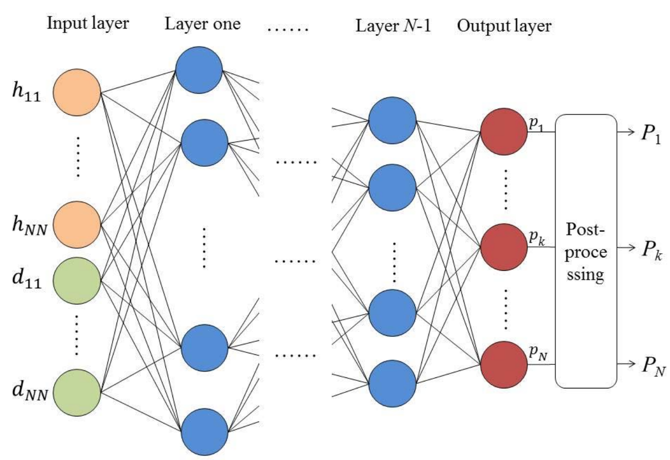

Fig. 2. The neural network architecture.

The kernel of modern ML discipline is the DNN. A DNN has multiple layers (Fig. 2). One of the most quintessential DNNs is the *feedforward neural network* (FNN). The intrinsic of FNN is validated by a fundamental theorem: UAT [8]. According to UAT, even a single layer of FNN can approximate almost all mathematical functions. The only issue is that a great number of units (neurons) is needed, incurring the high computational complexity. In the present work, we design an FNN with multiple layers. In the high level, we follow the general guidelines to construct the FNN. However, in several positions, we need to pay special attention to elaborate the generality. For example, our network should satisfy two constraints in eq. (3a) and eq. (3b) as well as maximization overall sum-rate.

To improve QoS satisfaction rate (QoS-SRate), a post-processing step is proposed. That is, at the output layer of the neural network, a SINR verification process is added. If the output power value (p_k) cannot satisfy the SINR request ($u_{k,min} \leq u_k$), then the power value will increase by 0.1 until it satisfies the SINR request ($P_{k,\text{min}qos}$) or reaches to the Pmax value ($P_{k,max}$).

$$P_k = \min(\max(p_k, P_{k,\text{min}qos}), P_{k,max}) \quad (5)$$

The post-processing will degrade to max power generate if all the output P_k equal to $P_{k,max}$. However, this is not happened in numerical experiments.

TABLE II. QoS SRate Test on Samples from Model-3

QoS (dB)	1	3	5	7	10	15	20	25
Fmincon	1	1	1	1	1	1	1	1
FNN_NoPP	0.94	0.82	0.75	0.72	0.68	0.64	0.62	0.60
FNN_PP	1.00	0.99	0.98	0.98	0.97	0.96	0.96	0.96

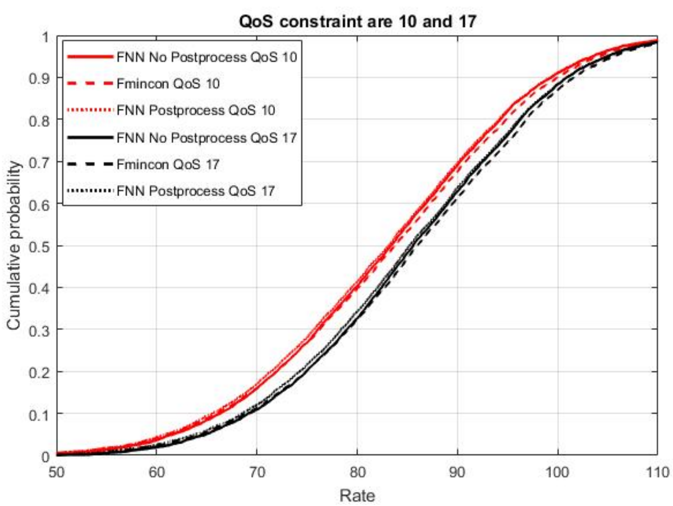

Fig. 3. Sum-rate with/without proposed post-processing on Model-3.

As can be observed from Table II, with the post-processing, the QoS satisfaction rates are largely improved. Fig. 3 shows the proposed post-processing slightly reduced the performance of the proposed FNN. However, the difference is very small compared its ability in improving the QoS satisfaction rate as shown in Table II. This indicates the effective of the proposed post-processing.

V. Experimental Results and Remarks

To verify the functionality and performance of the proposed optimization model, we conducted extensive numerical experiments. In the experimental setting, a standard optimization procedure, Fmincon in Matlab, is used to generate the training data. Fmincon is a well-known solver for the constrained *nonlinear programming* (NLP) problems. The main reason that we use Fmincon is that it is robust, and in most cases, it can converge to the satisfactory solution. In addition, it can be used to solve all the three optimization models described in Section III. Sometimes a few undesirable results could be generated from Fmincon, but these results can be easily identified and eliminated from the training samples. We use 80% of the training dataset as the *training samples*, while 20% as the *validation samples*.

We note that the excessive number of nodes is not desirable in most practical D2D deployments. Consequently, in the numerical experiments, we chose N = 10 pairs of transceivers. The following parameters are adopted: the radius = 1000 meters, Pmax = 21 dBm, and the AWGN power = -143.97 dBm. The layer size of the FNN is set to [200, 80, 80, 10]. We use the *scaled conjugate gradient descent* (SCGD) backpropagation method to update the FNN's weights values. The learning rate value is chosen as 0.01. For visualization purpose, the experimental performance is measured in terms of the *cumulative distribution function* (CDF). The CDF of all three models are computed based on the optimized power values of WSR in eq. (2).

A. Comparison of Models under Training Samples Generation

For the purpose of verifying the effectiveness of those three models described in Section III, the following two experiments were conducted. Note that Model-3 is the model optimized without constraint (3a).

First, the *successful optimization rate* (SOR) based on Fmincon toolbox is calculated and compared among these three models. In the current setting, the SOR measures the success proportion of Fmincon in 100 experimental trials. Given the same inputs, the SOR profiles for these three models are shown in Fig. 4. The goal here is to observe the impact of the model structure on the SOR. As shown in Fig. 4, Model-2 behaves best for SOR with different SINR requirements. On the other hand, the SOR of Model-3 is better than Model-1 in the regime of $u_{min} \leq 15$. However, in the large SINR regime, the performance of Model-3 gradually diminishes.

Second, the computational time of the three computational models are shown in Fig. 5. Here the computational time is the average time of 10 successful optimization trials. All tests are conducted with different QoS requirements $u_{k,\text{ min}}$. It is shown that both Model-2 and Model-3 have lower computational costs than Model-1. Apparently, Model-3 has the lowest computational cost. This implies that Model-3 would have the computational advantage for large datasets over the other two models.

From these experiments, it is observed that the proposed models (Model-2 and Model-3) perform better than the conventional WSR model (Model-1), with respect to both SOR and computational costs in generating training samples with Fmincon.

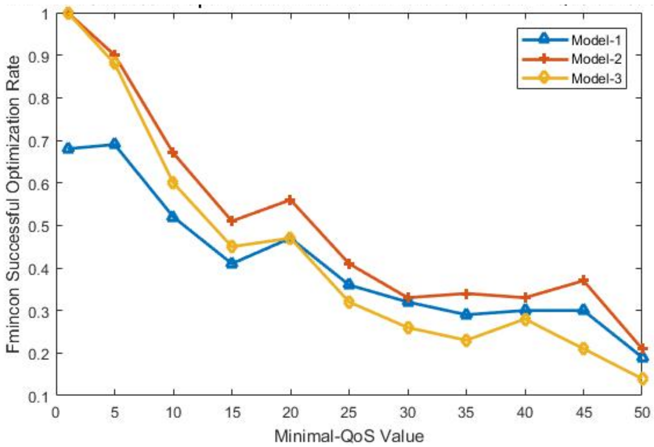

Fig. 4. Comparison the averaged power *P* under normal and infeasible cases.

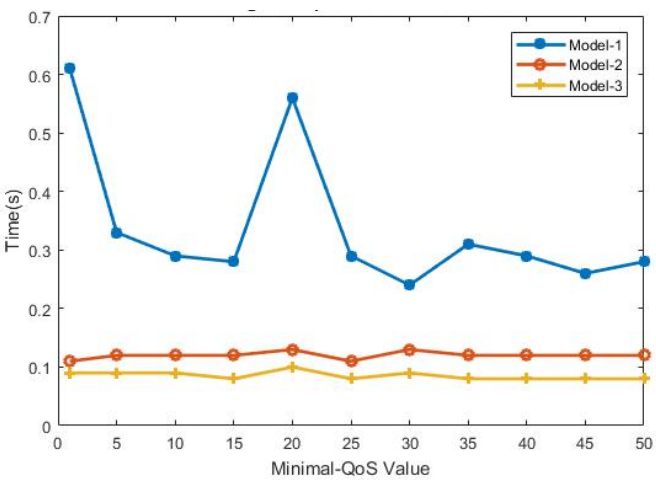

Fig. 5. Computational time comparison.

B. *Comparison of Models under FNN Optimization Performance*

In the training phase for the FNN, a wide range of QoS thresholds are used. We set six QoS thresholds with a wide range, i.e., 5 dB, 10 dB, 15 dB, 20 dB, 25 dB, and 30 dB. For each QoS threshold, 50,000 optimized samples are used as the targets for training, resulting in 300,000 samples in total. In the input side, each vector contains 100 channel gains and 100 link lengths. Thus, the dimension of each single vector is 200. Accordingly, the overall size of the input data for this FNN is 200×300,000. The numerical experiments are implemented in MATLAB R2017a on a computer with 12.0 GB RAM, Intel(R) i7 CPU 4.00GHz. The FNN is trained in the MATLAB *neural network toolbox* on the same computer. The same wireless network parameters are fed into Fmincon with three objective functions (Model-1, Model-2, Model-3), resulting three datasets of optimized powers. These datasets are used to train three different neural networks, i.e., FNN_m1, FNN_m2, and FNN_m3.

In the training process, when the total number of training epoch was set to 10,000 for all three models, then Model-3 and Model-2 converged respectively in 5 hours 57 minutes and 5 hours 50 minutes, with 0.033 and 0.038 as the best validation performance. However, Model-1 spent 7 hours 06 minutes with the best validation performance 0.043. This indicates that Model-3 is the easiest to learn with a neural network. The reason is that the optimization process of Model-3 without QoS constraint can be readily learned. In summary, this training process verified the superior performance of the proposed Model-3 over Model-1for training an FNN model.

Besides training, the standard procedure of FNN also includes a test stage. We conducted test experiments with different QoS constraints range from 1 dB to 40 dB with 1 dB interval. Each test set contains 5,000 samples. Under the QoS constraints, Fmincon generated three test datasets from three models, i.e., D1, D2, and D3 as ground truth. The final performance is measured and reported with respect to the WSR in eq. (2). The results of FNN are compared with random power generation method (RndP), as well as max power generation method (MaxP).

TABLE III. COMPARISION OF SUM-RATE ACCURACY ON THREE DATASETS

	QoS	1dB	9dB	17dB	25dB	33dB	40dB
	Fmincon	1.00	1.00	1.00	1.00	1.00	1.00
D1	FNN_m1	**0.98**	**0.98**	**0.98**	**0.99**	**0.99**	**0.99**
	FNN_m2	0.97	0.98	0.98	0.98	0.98	0.98
	FNN_m3	0.97	0.97	0.98	0.98	0.98	0.98
	RndP	0.93	0.94	0.95	0.95	0.95	0.95
	MaxP	0.91	0.92	0.92	0.92	0.93	0.93
D2	FNN_m1	**1.01**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	FNN_m2	1.00	1.00	1.00	1.00	1.00	1.00
	FNN_m3	1.00	0.99	0.99	0.99	0.99	1.00
	RndP	0.96	0.96	0.96	0.96	0.97	0.97
	MaxP	0.93	0.93	0.94	0.94	0.94	0.94
D3	FNN_m1	**1.01**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	FNN_m2	1.00	1.00	1.00	1.00	1.00	1.00
	FNN_m3	1.00	0.99	0.99	0.99	0.99	1.00
	RndP	0.96	0.96	0.96	0.96	0.97	0.97
	MaxP	0.93	0.93	0.94	0.94	0.94	0.94

Red, green, blue indicate the top 1, top 2, and top 3 best sum-rate accuracy under each test dataset.

The comparison with respect to sum-rate value of Fmincon, trained FNNs, random power, and max power with three test datasets are shown in Table III. As seen, the pre-trained FNNs achieve much better sum rate accuracy than the RndP and MaxP methods. With different QoS constraints and different test datasets, FNN_m1 gets the highest sum-rate accuracy. When the datasets are generated with Model-2 and Model-3 by Fmincon, FNN_m1 gets the better performance than Fmincon. This behavior indicates that Model-1 gains the robust performance for FNN optimization with respect to the sum rate accuracy. All three pre-trained FNNs achieved above 98% approximation performance. However, FNN_m1 gains better sum rate accuracy than the other two pre-trained FNNs through all test datasets. Note that in the training process only small part of the QoS constraints values are used to generate training samples. Therefore, the pre-trained FNNs have generation performance on both lower (QoS=1 dB) constraints and higher QoS (QoS=33 dB, 40 dB) constraints.

It should be mentioned that the trained FNN needs to not only learn to optimal the power values, but also needs to satisfy the QoS constraint. The comparisons with respect to QoS satisfaction rate (QoS-SRate), with FNN, random power, and max power averaged through three test datasets are shown in Table IV. As observed, with the post-processing step, all the trained FNNs achieve above 99% QoS-SRate, much better than the RndP and MaxP methods, especially with larger QoS constraints. This verifies the effectiveness of the post-processing in improving the QoS-SRate. In addition, the post-processing is independent from the sum-rate optimization process. Therefore, it is able to sustain high QoS-SRate with large QoS constraints, such as QoS equals to 40 dB.

TABLE IV. COMPARISION OF QOS-SRATE

QoS	1dB	9dB	17dB	25dB	33dB	40dB
Fmincon	1.00	1.00	1.00	1.00	1.00	1.00
RndP	0.83	0.50	0.42	0.37	0.36	0.33
MaxP	0.95	0.74	0.70	0.68	0.68	0.66
FNN_m1	1.00	1.00	1.00	1.00	1.00	1.00
FNN_m2	1.00	1.00	1.00	1.00	1.00	1.00
FNN_m3	1.00	1.00	1.00	1.00	1.00	1.00

C. Comparison of Models under Computational Costs

The overall computational time of different models and FNNs are measured on the same CPU platform, as illustrated in Table V. The computational times of Model-1, Model-2, and Model-3 were measured using Fmincon optimization procedure.

TABLE V. COMPARISION OF OVERALL COMPUTATIONAL TIME (S)

	QoS	1dB	9dB	17dB	25dB	33dB	40dB
Fmincon	Model-1	0.17	0.15	0.14	0.13	0.13	0.21
	Model-2	0.06	0.06	0.06	0.06	0.06	0.10
	Model-3	0.05	0.05	0.05	0.04	0.04	0.07
FNN	FNN_m1	0.01	0.01	0.01	0.01	0.01	0.01
	FNN_m2	0.01	0.01	0.01	0.01	0.01	0.01
	FNN_m3	0.01	0.01	0.01	0.01	0.01	0.01

As shown in Table V, Model-3 achieves the lowest computational cost compared to Model-1 and Model-2. This verifies the validity of the proposed model in generating training samples. However, the computational costs of Model-3 are still four to five times higher than the FNN-based optimization process. In addition, the computational time of FNN remains unchanged under different QoS constraints. These results verify the robustness and effectiveness of FNN in computational costs. It should be mentioned that the proposed post-processing took about 0.1E-4s on average. It added little computational costs on the overall FNN framework.

VI. CONCLUSION

Resource optimization for small-cell wireless networks is more complicated than the conventional networks due to their ad hoc nature. The solution is also needed quickly due to the highly dynamic yet random temporal and spatial variations. In this work, we focus on a D2D network and apply the ML method to solve the power optimization problem. It is shown that the ML method has achieved several benchmarks in terms of the training process, sum-rate accuracy, QoS-SRate, computational efficiency, etc. under different mathematical optimization models. It is expected the demonstrated insights will inspire further interests in applied ML discipline.

REFERENCES

[1] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, “Machine learning paradigms for next-generation wireless networks,” *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98-105, 2017.

[2] C.-X. Wang, F. Haider, X. Gao, X.-H. You, Y. Yang, D. Yuan, H. M. Aggoune, H. Haas, S. Fletcher, and E. Hepsaydir, “Cellular architecture and key technologies for 5G wireless communication networks,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 122-130, 2014.

[3] N. Lee, X. Lin, J. G. Andrews, and R. W. Heath, “Power control for D2D underlaid cellular networks: Modeling, algorithms, and analysis,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 1, pp. 1-13, 2015.

[4] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, “Learning to optimize: Training deep neural networks for interference management,” *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438-5453, 2018.

[5] W. Lee, M. Kim, and D.-H. Cho, “Deep power control: Transmit power control scheme based on convolutional neural network,” *IEEE Communications Letters*, vol. 22, no. 6, pp. 1276-1279, 2018.

[6] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, “A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs,” in *Proc. of IEEE ICC*, 2017, pp. 1-6.

[7] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, “Learning to optimize: Training deep neural networks for wireless resource management,” in *Proc. of IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications* (SPAWC), 2017, pp. 1-6.

[8] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989.

[9] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, “An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel,” *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331-4340, 2011.

[10] A. Zappone, M. Debbah, and Z. Altman, “Online energy-efficient power control in wireless networks by deep neural networks,” in *Proc. of IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications* (SPAWC), 2018, pp. 1-5.

[11] F. Liang, C. Shen, W. Yu, and F. Wu, “Towards optimal power control via ensembling deep neural networks,” arXiv preprint arXiv:1807.10025, 2018.

[12] M. Eisen, C. Zhang, L. Chamon, D. D. Lee, and A. R. Ribeiro, “Learning optimal resource allocations in wireless systems,” *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2775-2790, 2019.

[13] J. Zhou, X. Liu, Y. Tao, and S. Yu, “QoS-aware power management with deep learning,” in *Proc. of IFIP/IEEE International Symposium on Integrated Network Management*, 2019.