Energy Efficiency Optimization in LoRa Networks - A Deep Learning Approach

Lam-Thanh Tu, Abbas Bradai, Ben Ahmed, Sahil Garg, Yannis Pousset

Energy Efficiency Optimization in LoRa Networks - A Deep Learning Approach

Lam-Thanh Tu, Abbas Bradai, Olfa Ben Ahmed, Sahil Garg and Yannis Pousset

***Abstract*— The optimal transmit power that maximizes energy efficiency (EE) in LoRa networks is investigated by using deep learning (DL) approach. Particularly, the proposed artificial neural networks (ANNs) is trained two times; in the first phase, the ANNs is trained by the model-based data which are generated from the simplified system model while in the second phase, the pre-trained ANNs is re-trained by the practical data. Numerical results show that the proposed approach outperforms the conventional one which directly trains with the practical data. Moreover, the performance of the proposed ANNs under both partial and full optimum architecture are studied. The results depict that the gap between these architecture is negligible. Finally, our findings also illustrate that instead of fully re-trained the ANNs in the second training phase, freezing some layers are also feasible since it does not significantly decrease the performance of the ANNs.**

***Index Terms*— Energy Efficiency, LoRa Networks, Stochastic Geometry, Deep Learning, Poisson Cluster Process.**

I. Introduction

Energy efficiency (EE) is one of the long lasting problems in wireless communications systems. However, in the past, the network operator/planning primarily focused on maximizing the spectral efficiency (SE) as well as enhancing the coverage area, as a result, the maximum transmit power was typically considered as one of the optimal solutions. Nonetheless, such the approach, of course, reduces dramatically the energy efficiency. The issue is even more serious since it is expected that by 2022 there will be 12.3 billion end-devices (EDs) to be connected to the wireless networks [1] and making the internet-of-things (IoTs) becomes feasible. As a consequence, minimizing the power consumption or maximizing the EE while guaranteeing the SE has been emerged as one of the most important issues in wireless networks. Despite the fact that the operation of the base stations (BSs) has been optimized to maximize the EE of the cellular networks [2], super-dense deployment makes its too bulky, thus, letting its less attractable for supporting such a massive low power networks. Fortunately, the low power wide area network (LPWAN) is regarded as one of the most suitable technologies thanks to its properties, i.e., low power consumption, low cost and wide coverage area [3]. Among all the available LPWAN techniques, i.e., SigFox, Weightless, NB-IoTs, etc., LoRa is gained lots of attraction in both academia and industry. By actively fine-tuning its parameters, i.e., the spreading factor (SF), the coding rate (CR) and the bandwidth (BW), LoRa is able to serve a wide range of IoTs applications/devices, e.g., e-Health, smart city, smart home, that, in general, have different quality-of service (QoS).

Deep learning (DL), on the other hand, is proven itself as one of the best ones among all machine learning (ML) techniques when a large number of data are available. Nevertheless, unlike other domains where the mathematical modelling is primarily difficult to be performed and the pure data-driven approach like deep learning seems to be the sole solution, wireless communications, on the contrary, have always depended on a robust mathematical modelling for system design, analyzing, optimization and can be considered as model-based approach. Nonetheless, as the dramatic evolution of the wireless networks, i.e., the exponential growth of end-devices with different applications are connected to wireless networks, making its more complex, hence, the mathematical modelling is steadily losing its accuracy as well as mathematical tractability. As a result, deep learning has recently been commenced applying into wireless communications. The application of DL in wireless networks, differently, does not mean that the prior mathematical modelling is ignored. In fact, the most feasible solution of applying DL in wireless communications is to combine the advantages of both data-driven and model-based approach [4]. The main target of this combination is to synergistically exploit the deep and expert knowledge from theoretical models even though it may be inaccuracy and cumbersomeness to facilitate the use of DL in wireless networks [5]. This approach can also be considered as transfer learning [6], where the artificial neural networks (ANNs) is first trained by faults data and then re-trained by the empirical data. Of course, such the approach requires a strong mathematical modelling which is as close as possible to the empirical data in order to attain the highest performance. As a result, in this paper, we maximize the energy efficiency in LoRa networks by combining the model-based and data-driven approach.

The performance of the energy efficiency in LoRa networks were studied in [7], [8] In [7], the energy efficiency was investigated by considering other medium accesses rather than pure ALOHA. By jointly exploiting user scheduling and SF assignment, the maximization of system EE was investigated in [8]. In spite of the use of DL has recently been received tremendous attention in cellular networks, its application in LoRa networks is still in infancy stage. There were a few works which applied ML/DL in LoRa networks, [9], [10]. Particularly, in [9], the process of networks configuration was formulated as a rerinforcement learning (RL) problem. The time difference of arrival positioning method in LoRa networks

L.-T. Tu, A. Bradai, O. Ben Ahmed and Y. Pousset are with the Institute XLIM, University of Poitiers, France (email: lam.thanh.tu@univ-poitiers.fr; abbas.bradai@univ-poitiers.fr; olfa.ben.ahmed@univ-poitiers.fr; yannis.pousset@univ-poitiers.fr).

S. Garg is with Ecole de technologie superieure, Universite du Quebec, Montreal (email: sahil.garg@ieee.org).

was improved by applying deep learning in [10].

In the present work, different to these above-mentioned papers, we maximize the energy efficiency in LoRa networks respect to the transmit power under practical scenarios where the distribution of the end-devices and the small-scale fading follow general distributions, i.e., the Poisson cluster processes (PCP) and the Nakagami-m distribution, combined with the imperfect power consumption at the end-devices. Of course, it is crystal clear that the optimal transmit power that maximizes the EE under such the general system model can not be attained based on the mathematical frameworks, thus, the deep learning approach is utilized instead. Our proposed DL approach (called double training approach), however, requires a relatively small practical data to attain the high accuracy output and is different to the conventional approach (called direct training approach) where a large number of practical data is imperatively needed in order to fully train the ANNs. To realize such the well-trained neural networks with a relative small practical data, the proposed artificial neural networks is trained two times. In the first training phase or pre-trained phase, the proposed ANNs is trained based on the model-based data and in the second training phase, the pre-trained ANNs is re-trained based on the practical data. Here, the model-based data in the first training phase are generated based on the mathematical frameworks under the simplified system model that may be inaccuracy but may provide some expert knowledge and can be helpful for the second training; the practical data in the second training phase, diversely, are uniquely generated by Monte Carlo simulation that is resources-consuming compared to the mathematical frameworks. To be more specific, the main contributions and novelties of this paper are summarized as follows: i) the distribution of the end-devices and the small-scale fading follow Poisson cluster processes and Nakagami-m distribution, respectively; the power consumption at EDs is impaired by an additive noise that follows either Gaussian or uniform distribution; ii) the closed-formed expression of the energy efficiency under the simplified system model is provided; iii) the first training set is generated by deploying the mathematical frameworks under the simplified system model. Particularly, the optimal transmit power in this phase is attained by numerical solving a non-linear equation; iv) the proposed double training approach outperforms the direct training approach under all considered metrics, i.e., the mean square error (MSE) and R squared, $\mathcal{R}^2$, or the coefficient of determination; v) our findings show that increasing training data monotonically ameliorates the performance of the neural networks. Nonetheless, enhancing the ANNs performance by raising either the number of epochs or the number of neurons does not always benefit; vi) the performance of the ANNs under various optimal network architecture, i.e., partial and full optimum architecture, are investigated. The findings depict that the gap between these optimum architecture is negligible, thus, the partial optimum architecture is preferable provided that the resources consumption are taken into account; and vii) our findings also illustrate that it is feasible to freezing some layers during the second training phase which does not significantly affect the performance of the ANNs.

The rest of this paper is organized as follows. In Section II, the considered system model is presented. In Section III, we formulate the optimization problem and the design of the proposed neural networks is provided in Section IV. In Section V, the performance of the proposed ANNs is evaluated and discussed under various scenarios. Finally, Section VI concludes the paper.

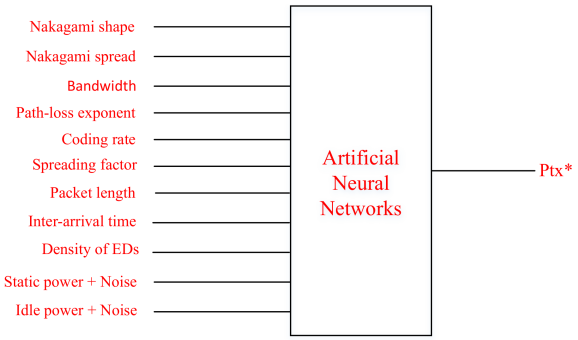

Fig. 1. Schematic of the considered artificial neural networks

Notations: $\Pr(.)$ and $\mathbb{E}\{.\}$ are the probability and the expectation operators; $\max\{.\}$, $\min\{.\}$ and $\exp(.)$ being the maximum, minimum and exponential function; uppercase boldface letters for vectors; $|X|$ is the size of set X; $\mathbf{1}(x)$ is the indicator function which is equal to 1 if $x > 0$ and 0 otherwise; $\log(.)$ is the logarithm function; $F_X(x)$ and $f_X(x)$ being the cumulative distribution function (CDF) and the probability density function (PDF) of random variable (RV) X; $\dot{f}(x)$ is the first-order derivative of function f over x, $\dot{f}(x) = df(x)/dx$.

II. System Model

A. LoRa Networks Modeling

Let us consider an uplink single gateway LoRa networks where the gateway is located at the center of the disc with area $A = \pi R^2$, here, R is the network radius and a number of EDs which follow Poisson cluster processes (or doubly Poisson point processes), with density $\lambda_{\mathcal{C}} > 0$, inside A and $\lambda_{\mathcal{C}} = 0$, otherwise. In particular, in this work we consider two notable Poisson cluster processes, i.e., the Matérn and the Thomas cluster process with corresponding density $\lambda_{\mathrm{Mat}} = \lambda_P \overline{C}_{\mathrm{Mat}}$ and $\lambda_{\mathrm{Tho}} = \lambda_P \overline{C}_{\mathrm{Tho}}$, where λ_P is the density of the parent point process and $\overline{C}_u$, $u \in \{\mathrm{Mat}, \mathrm{Tho}\}$, is the average number of offspring per cluster in the u cluster process [11]. The offspring of the Matérn cluster process are independently and uniformly distributed in a disc of radius r_{Mat} around the parent point while the offspring of the Thomas cluster process are scattered with variance σ^2_{Tho} around each parent point.

B. Channel Modelling

Let us consider a generic signal from an arbitrary ED to the gateway, it is impaired by both the small-scale fading and large-scale path-loss. It should be emphasized that the impact of the shadowing is implicitly studied by modifying the density of the EDs [2].

1) Small-scale fading: Let us denote h_q is the fading from an arbitrary node q to the gateway which follows Nakagami-m distribution with corresponding shape and spread parameters, i.e., $m \geq 1/2$ and Ω. There is no doubt that Nakagami-m fading is one of the most general fading distributions that can represent other well-known distributions by properly adjusting its shape parameter, e.g., $m = 1$, is the Rayleigh fading and

$m = 1/2$, is the single-sided Gaussian distribution. In addition, assuming that time is slotted (slotted ALOHA medium access [7]) and the fading remains constant during one time-slot and changes between time-slot.

2) Large-scale path-loss: Consider a transmission link from a generic node q to the gateway, the large-scale path-loss is formulated as

$$\rho_q = K_0 r_q^{\beta}, \tag{1}$$

where $\beta > 2$ and $K_0 = (4\pi f_c/c)^2$ are the path-loss exponent and the path-loss constant, respectively; f_c is the carrier frequency and $c = 3 \times 10^8$ (in meter per second) is the speed of light; and r_q is the distance from node q to the gateway.

C. Power Consumption Modelling

Assuming that the ED is operated either in transmission mode or sleep mode. To be more precise, the ED is considered in transmission mode providing that it transmits packets to the gateway and in sleep mode otherwise. The power consumption under transmission mode comprises of two parts, i.e., the transmit power, P_{tx}, and the static (circuit) power, P_{sta}; while in the idle mode, the ED consumes P_{idl} power. Here, the static power consumption comprises of all other power consumption excluding the transmit power in the transmission mode, i.e., wake up, radio preparation, wait and receive the 1st and 2nd windows, radio off and so on. Furthermore, we also consider the impact of the hardware impairment on the static and idle power; indeed, the static and idle power are impaired by an additive noise which follows either Gaussian or uniform distribution. As a result, the practical static and idle power denoted as $\widetilde{P}_{\text{sta}}$ and $\widetilde{P}_{\text{idl}}$ are written as follows:

$$\widetilde{P}_{\text{sta}} = P_{\text{sta}} + \omega_{\text{sta}}, \quad \widetilde{P}_{\text{idl}} = P_{\text{idl}} + \omega_{\text{idl}}, \tag{2}$$

where ω_x, $x \in \{\text{sta, idl}\}$, is the random variable (RV) and P_{sta} and P_{idl} are the ideal static and idle power.

III. Problem Formulation

In this work, the principal objective is to identify the optimal transmit power denoted as P_{tx}^* that maximizes the energy efficiency (in bits/Joule) of whole networks given a set of input parameters, i.e., the fading parameters included both shape and spread factor, the transmission bandwidth, the spreading factor, the coding rate, the inter-arrival time between two packets, T_{in}, the packet length, L_{pac}, the path-loss exponent, the density of the PCP, $\lambda_{\mathcal{C}}$, and the imperfection static and idle power as shown in Fig. 1. Particularly, the problem in Fig. 1 can be formulated as follows:

$$\max_{P_{\text{tx}} \in \left[P_{\text{tx}}^{\min}, P_{\text{tx}}^{\max}\right]} \text{EE} = \frac{\text{PSE}}{\text{Pcon}} = \frac{\lambda_{\mathcal{A}} \text{BW} \log_2(1+\gamma_{\text{D}}) \text{Pcov}(\gamma_{\text{D}})}{\lambda_{\mathcal{A}}\left(P_{\text{tx}} + \overline{P}_{\text{sta}}\right) + (\lambda_{\mathcal{C}} - \lambda_{\mathcal{A}})\overline{P}_{\text{idl}}}. \tag{3}$$

It is noted that the range of the transmit power in (3), without loss of generality, can go from zero to infinity, i.e., $P_{\text{tx}}^{\min} \to 0$ and $P_{\text{tx}}^{\max} \to \infty$; where PSE refers to the potential spectral efficiency (in bits/s/m^2) that measures number of bit successfully transmitted per unit area and is formulated as $\lambda_{\mathcal{A}} \text{BW} \log_2(1+\gamma_{\text{D}}) \, \text{Pcov}(\gamma_{\text{D}})$; $\text{Pcov}(\gamma_{\text{D}})$ is the coverage probability of an arbitrary link and formulated as

$$\text{Pcov}(\gamma_{\text{D}}) = \Pr\left\{ \text{SIR} = \frac{S}{I_S} \geq \gamma_{\text{I}}, \text{SNR} = \frac{P_{\text{tx}} S}{\sigma^2} \geq \gamma_{\text{D}} \right\}, \tag{4}$$

where SIR is signal-to-interference ratio and is computed as $\text{SIR} = \frac{P_{\text{tx}} S_0}{P_{\text{tx}} I_S} = \frac{P_{\text{tx}} h_0^2/\rho_0}{P_{\text{tx}} \sum_{i \in \Phi^{\mathcal{A}} \setminus (0)} h_i^2/\rho_i}$; S_0 is the signals from the ED of interest to the gateway; I_S is the aggregate interference from all active EDs except for the desired one; assuming that packets with different spreading factor are perfectly orthogonal, hence, there is no inter spreading factor interference at the gateway [12]; h_0^2 and ρ_0 being the channel gain and large-scale path-loss of the ED of interest while h_i^2 and ρ_i are the channel gain and the large-scale path-loss from interferer i to the gateway; $\Phi^{\mathcal{A}} \setminus (0)$ is the set of active EDs except for the desired ED which density $\lambda^{\mathcal{A}} = p_A \lambda_{\mathcal{C}}$ under the considered area; where $p_A = \frac{1}{T_{\text{in}}} \frac{L_{\text{pac}}}{R_{\text{bit}}}$ being active probability. In this work, assuming that the length of the packet, L_{pac}, is identical among all EDs and R_{bit} is the bit rate which is computed as $R_{\text{bit}} = \text{SF} \frac{\text{BW}}{2^{\text{SF}}} \text{CR}$ [3]. SNR is the signal-to-noise ratio and is formulated as $\text{SNR} = P_{\text{tx}} S_0 / \sigma^2$ where $\sigma^2 = 10^{(-174 + \text{NF} + 10 \log_{10} \text{BW})/10}$ [12] is the noise variance of the AWGN noise; NF is the noise figure (in dBm) at receiver. In (3), Pcon is the average power consumption of the whole networks measured in Watt/m^2; the first term of Pcon, $\lambda_{\mathcal{A}} \left(P_{\text{tx}} + \overline{P}_{\text{sta}}\right)$, accounts for the power consumption under transmission mode and the remain term, $(\lambda_{\mathcal{C}} - \lambda_{\mathcal{A}}) \overline{P}_{\text{idl}}$ is under idle mode; where $\overline{P}_x = P_x + \mathbb{E}\{\omega_x\}$, $x \in \{\text{sta,idl}\}$, is the average dissipation of the static and idle power.

It is apparent that the most intuitive approach to solve the optimization problem in (3) is to compute the EE in closed-form expression followed by solving a non-linear equation to obtain P_{tx}^*. However, as shown in the sequel, it is impossible to obtain the closed-form expression of the EE under the considered system model. Moreover, even the distribution of the distance from an arbitrary node to the gateway as well as the aggregate interference are also unfeasible to represent in the closed-form expressions. Especially, by direct inspection (3), it is evident that the framework of the Pcov is essential in order to compute the EE, let us formulate the Pcov as follows:

$$\text{Pcov}(\gamma_{\text{D}}) = \int_{s = \sigma^2 \gamma_{\text{D}}/P_{\text{tx}}}^{\infty} F_{I_S}(s/\gamma_{\text{I}}) \, f_S(s) \, ds, \tag{5}$$

where $f_S(s)$ and $F_{I_S}(x)$ are the PDF of the intended signal and the CDF of the aggregate interference. Nonetheless, the CDF of I_S has never existed even with the simplest scenario, i.e., the EDs follow Poisson point process (PPP); the PDF of the intended signal, on the contrary, can be computed in the closed-form expression under some special cases, for example, the EDs follow PPP combined with Rayleigh fading and special value of the path-loss exponent. It, as a result, is unworkable to obtain the closed-form expression of both Pcov and EE even with the simplest case [13].

As a result, in this paper, we are going to find the P_{tx}^* by using deep learning approach. Nevertheless, different to the conventional DL approach that requires enormous training data to fully train the ANNs or simply called the data-driven approach. In this manuscript, we synergistically combine the model-based with the data-driven approach to derive the optimal transmit power in (3). To do so, we split the training process into two phases instead of one as the conventional approach. In the first training phase, the ANNs is trained by

$$\widetilde{P}\left(\gamma_{\mathrm{D}}\right)=\mathcal{M}^{-1}(\gamma_{\mathrm{I}})^{-\alpha}\left(1-\exp\left(-\mathcal{M}\min\left\{(\mathcal{A}\gamma_{\mathrm{I}})^{\alpha},1\right\}\right)\right)+\exp\left(-\mathcal{M}\right)\left(\min\left\{\mathcal{A}^{\alpha},1\right\}-(\gamma_{\mathrm{I}})^{-\alpha}\right)\mathbf{1}\left(\mathcal{A}\gamma_{\mathrm{I}}-1\right) \quad (6)$$

$$\dot{\widetilde{P}}\left(x\right)=\dot{\mathcal{A}}\left(x\right)\alpha(\mathcal{A}\left(x\right))^{\alpha-1}\left(\exp\left(-\mathcal{M}(\mathcal{A}\left(x\right)\gamma_{\mathrm{I}})^{\alpha}\right)\mathbf{1}\left(1-\mathcal{A}\left(x\right)\gamma_{\mathrm{I}}\right)+\exp\left(-\mathcal{M}\right)\mathbf{1}\left(\mathcal{A}\left(x\right)\gamma_{\mathrm{I}}-1\right)\mathbf{1}\left(1-\mathcal{A}\left(x\right)\right)\right) \quad (8)$$

the model-based data which may be simplified and inaccuracy but can bring some expert knowledge and useful for the second phase. In the second training phase, the pre-trained ANNs is re-trained by utilizing a small amount of practical data which are generated via Monte Carlo simulation. In next section, the design of the proposed neural networks as well as the detail description of the training process are provided.

IV. Design of the neural networks

In this section, the proposed ANNs networks is designed from creating the data to selecting the optimal architecture of the ANNs as well as identifying the performance metrics which are used to evaluate the performance of the ANNs.

A. Generate data set

1) First training phase (pre-train phase) data set: It is obvious that data is the most important element of any neural networks. Under the considered system model, it is unreasonable to obtain a large number of data from either Monte Carlo simulations due to the resources constraint or the frameworks owing to the mathematical intractability as discussed in Section III. Consequently, in this section, we are going to simplify the system model in Section II so that the closed-form expression of the EE and the optimal transmit power, P_{tx}^{*}, can be numerically obtained.

Particularly, the simplified system model is described as follows: i) assuming that the ED is no longer followed PCP but PPP with density λ_P; ii) the ideal static and idle power at EDs; and iii) the aggregate interference is approximated by the dominant interferer [12] and the instantaneous fading by its average [13]. Other assumptions are remained the same as Section II. Based on these assumptions, the approximated coverage probability denoted as $\widetilde{P}\left(\gamma_{\mathrm{D}}\right)$ is computed in (6) at the top of this page. Here, $\alpha=2/\beta$, $\mathcal{M}=p_A\lambda_P\pi R^2$, $\mathcal{A}=\frac{m\theta P_{\mathrm{tx}}}{R^{\beta}K_0\sigma^2\gamma_{\mathrm{D}}}$ and $\mathbf{1}\left(x\right)$ is the indicator function.

Having the approximated Pcov in hand, the optimal transmit power under the simplified system model denoted as $\widetilde{P}_{\mathrm{tx}}^{*,1}$, is obtained by solving following non linear equation:

$$\dot{\widetilde{P}}\left(P_{\mathrm{tx}}\right)Q\left(P_{\mathrm{tx}}\right)-\widetilde{P}\left(P_{\mathrm{tx}}\right)\dot{Q}\left(P_{\mathrm{tx}}\right)=0, \quad (7)$$

where $\dot{\widetilde{P}}\left(x=P_{\mathrm{tx}}\right)$ is the first-order derivative of the approximated Pcov respect to the transmit power and is computed in (8) at the top of this page and $\dot{Q}\left(x\right)=p_A\lambda_P$ is the first-order derivative of the average power consumption respect to P_{tx}; $Q\left(x\right)=\lambda_{\mathcal{A}}\left(P_{\mathrm{tx}}+\overline{P}_{\mathrm{sta}}\right)+\left(\lambda_{\mathcal{C}}-\lambda_{\mathcal{A}}\right)\overline{P}_{\mathrm{idl}}$ and $\dot{\mathcal{A}}\left(x=P_{\mathrm{tx}}\right)=d\mathcal{A}/dx=\frac{m\theta}{R^{\beta}K_0\sigma^2\gamma_{\mathrm{D}}}$, respectively.

2) Second training phase (re-train phase) data set: The data in this phase are created imperatively by employing Monte Carlo simulation. However, owing to the resources constraint, a limited amount of data are created compared to the previous phase.

B. Data normalization

The data set is normalized before putting into the ANNs. In this work, the simple max-min normalization is applied as follows:

$$x=\frac{x_{\mathrm{ori}}-\min\left(x_{\mathrm{ori}}\right)}{\max\left(x_{\mathrm{ori}}\right)-\min\left(x_{\mathrm{ori}}\right)}, \quad (9)$$

where x_{ori} and $x\in[0,1]$ are the input and output of the normalization process. In the sequel, without explicit explanation, we assume that the data set has already been normalized.

C. Networks Architecture

To maximize the performance of the ANNs, optimizing the networks architecture, i.e., the optimal number of hidden layers and/or number of neurons of each layer, is essential along with other hyper-parameters optimization. As the considered ANNs will be trained two times; it, in theory, exists two optimal typologies, the first one which merely optimizes from the 1st training set and another which optimizes from both the 1st and 2nd training set. In mathematics, the optimal architecture based on the 1st training set (partial optimum) and from both 1st and 2nd training set (full optimum) denoted as N_1^{*} and N_{1+2}^{*} as a function of the number of hidden layers, $\mathcal{L}\in\{\mathcal{L}_{\min},\ldots,\mathcal{L}_{\max}\}$, the number of neurons of all hidden layers are formulated as follows:

$$\begin{aligned}N_1^{*}\left(\mathcal{L}_1^{*},\mathbf{N}_1^{*}\right)&=\min\ \ L\left(\mathcal{L},\mathbf{N}|\Psi^{(1)}\right)\\ N_{1+2}^{*}\left(\mathcal{L}_{1+2}^{*},\mathbf{N}_{1+2}^{*}\right)&=\min\ \ L\left(\mathcal{L},\mathbf{N}|\Psi^{(1)}+\Psi^{(2)}\right),\end{aligned} \quad (10)$$

where $\mathbf{N}=[\mathcal{N}_1,\ldots,\mathcal{N}_{\mathcal{L}}]$ being a vector which contains number of neurons of all hidden layers; $\mathcal{N}_l\in\{\mathcal{N}_{\min},\ldots,\mathcal{N}_{\max}\}$, $l\in\{1,\ldots,\mathcal{L}\}$, is the number of neurons in l-th hidden layer; $\Psi^{(u)}$, $u\in\{1,2\}$, is the training set of u phase; $L\left(.\right)$ is the objective function which needs to be minimized; $(\mathcal{L}_a^{*},\mathbf{N}_a^{*})$, $a\in\{1,\ 1{+}2\}$, is the optimal solution based on u training set; $\{\mathcal{L}_{\min},\ldots,\mathcal{L}_{\max}\}$ and $\{\mathcal{N}_{\min},\ldots,\mathcal{N}_{\max}\}$ are the minimum and maximum of number of hidden layers and number of neurons of each layer, respectively.

It is evident that optimizing DL topology in (10) is always a cumbersome task as it requires not only the skills and experience of the user, but also the mastery of the data features [14]. As a result, the grid search method is applied to find out the optimal architecture with the help of [15]. In addition, to simplify the search space, we further assume that the number of neurons of all hidden layers is identical, i.e., $\mathbf{N}=\mathcal{N}$. Thus, the optimization problem in (10) can be re-written as follows:

$$\begin{aligned}N_1^{*}\left(\mathcal{L}_1^{*},\mathcal{N}_1^{*}\right)&=\min\ \ L\left(\mathcal{L},\mathcal{N}|\Psi^{(1)}\right)\\ N_{1+2}^{*}\left(\mathcal{L}_{1+2}^{*},\mathcal{N}_{1+2}^{*}\right)&=\min\ \ L\left(\mathcal{L},\mathcal{N}|\Psi^{(1)}+\Psi^{(2)}\right),\end{aligned} \quad (11)$$

In fact, the search space has massively reduced from $\mathcal{T}=\sum_{i=1}^{(\mathcal{L}_{\max}-\mathcal{L}_{\min})}\left(\mathcal{N}_{\max}-\mathcal{N}_{\min}\right)^{i}$ in (10) to $\mathcal{T}=\left(\mathcal{L}_{\max}-\mathcal{L}_{\min}\right)\left(\mathcal{N}_{\max}-\mathcal{N}_{\min}\right)$ in (11).

Remark 1: It is apparent that the full optimum will theoretically provide better performance compared to the partial optimum. Thus, the main purpose is to clarify the gap between the partial optimum versus the full optimum. In case the gap is relatively small, it is more beneficial to employ the partial optimum than its counterpart provided that the consumed resources are taken into consideration.

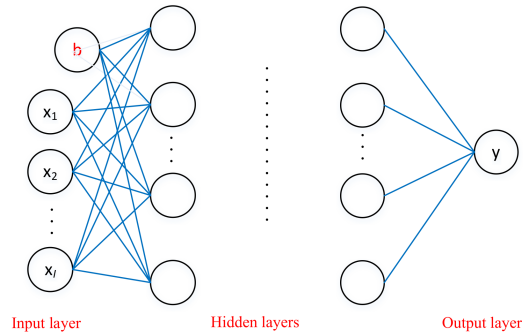

Fig. 2. The fully-connected feedforward neural networks

D. Loss function

As the considered problem belongs to the regression one, the typical mean square error is deployed as the loss function of the ANNs and is formulated as follows:

$$\min \quad \text{MSE} = (|\Psi|)^{-1} \sum\nolimits_{j\in\Psi} \left(y_j - \widehat{y}_j\right)^2, \tag{12}$$

where Ψ and $|\Psi|$ are the training set and its size; y is the observed output and $\widehat{y}$ is the predicted output.

E. Performance Metric

The mean square error and the R squared, $\mathcal{R}^2$, or the coefficient of determination are utilized as the main metrics to evaluate the performance of the ANNs. As the MSE measures the average squared difference between the estimated and true values and is provided in (12); the $\mathcal{R}^2$, differently, measures how close of two sets of data in terms of the distribution. Mathematical speaking, $\mathcal{R}^2$ can be formulated as follows:

$$\mathcal{R}^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} = 1 - \frac{\sum_{j\in\Psi}\left(y_j - \widehat{y}_j\right)^2}{\sum_{j\in\Psi}\left(y_j - \overline{y}\right)^2}, \tag{13}$$

where SS_{res} and SS_{tot} are the residual sum of squares and the total sum of squares. Here, SS_{res} measures the amount of variability that is left unexplained after performing the regression while SS_{tot} measures the total variance. As a consequence, if R squared towards 1, it explicitly means that most of the variability are explained by the model; $\overline{y} = (|\Psi|)^{-1}\sum_{j\in\Psi} y_j$ is the mean of the observed output.

F. Training process

Considering a fully-connected feedforward neural networks with $\mathcal{L}$ hidden layers, $\mathcal{N}$ neurons per hidden layer, one input layer with $\mathcal{I}$ input and single output as shown in Fig. 2. The input of the ANNs are the path-loss exponent, the fading parameters, the bandwidth, the coding rate, the spreading factor, the packet length, the inter-arrival time, the density of EDs, the practical static and idle power. The single output is the optimal transmit power, P^*_{tx}, which is obtained by (7) in the pre-trained phase or Monte Carlo simulation in the re-trained phase.

Given a training set, $\Psi^{(u)}$, $u \in \{1, 2\}$, the training process is commenced with normalizing the training set by using (9), followed by forward propagation. Particularly, let us denote x^l_n as the output signal at neuron n, $n \in \{1, \ldots, \mathcal{N}\}$, in hidden layer l, $l \in \{1, \ldots, \mathcal{L}\}$, and is computed as

$$x^l_n = \zeta\left(\sum_{i=1}^{\mathcal{V}} w^l_{i,n} x^{l-1}_i + b^l_n\right), \mathcal{V} = \begin{cases} \mathcal{V} = \mathcal{I}, & l = 1 \\ \mathcal{V} = \mathcal{N}, l \neq 1 \end{cases}, \tag{14}$$

where $w^l_{i,n}$ is the weight from node i in layer $l-1$ to node n in layer l; x^{l-1}_i is output of neuron i in layer $l-1$; when $l = 1$, we have x^0_i, $i \in \{1, \ldots, \mathcal{I}\}$, is the i-th input of the ANNs; $\zeta()$ is the activation function; b^l_n denotes the bias of node n in layer l. In this work, we utilize the sigmoid activation function as the training set is normalized to $[0, 1]$. The predicted output of the ANNs denoted as $\widehat{y}$ is linearly combined the output of the last hidden layer with the weight from the last hidden layer to the output as $\widehat{y} = \sum_{i=1}^{\mathcal{N}+1} w^{\mathcal{L}+1}_i x^{\mathcal{L}}_i + b_y$, b_y is the bias of the output. The forward pass ends by computing the loss function given in (12). The next step is backward propagation where we compute the gradient of the loss function respect to all the weights and bias of the networks. This gradient is then used to update the weights and bias to minimize the loss function by deploying Adam algorithm with adaptive learning rate [16] and we finish one training epoch. The training process is repeated by re-computing the forward and backward propagation; updating the loss function, weights and bias until reaching the stopping conditions, i.e., the maximum epoch or the MSE is smaller than a predefined threshold or the MSE starts increasing, etc.

Remark 2: In the step back propagation during the second training phase, it is unnecessary to update all the weights and bias. In fact, the ANNs can re-train partially instead of fully re-trained or the weight and bias of some layers can be kept constant during the second training phase.

V. Numerical Results and Discussion

In this section, numerical results are provided to confirm our findings. Unless otherwise stated, following values are used to generate the input training set in the first and second phases: $\beta \in [2, 6]$, $m \in [0.5, 30]$, $\Omega \in [1, 50]$, SF $\in \{7, \ldots, 12\}$, BW $\in \{1, \ldots, 500\}$ KHz, CR $= 4/(4+o)$, $o \in \{1, \ldots, 4\}$, $L_{\text{pac}} \in \{1, \ldots, 120\}$ (bytes), $T_{\text{in}} \in \{1, \ldots, 90\}$ (minutes), $P_{\text{cir}} \in [0.5, 5]$ (dBm), $P_{\text{idle}} \in [-10, -5]$ (dBm), $\lambda_P = \overline{N}/\pi R^2$ where $\overline{N} \in \{500, \ldots, 100000\}$ and R = 2000 m; γ_{I} = 1 dBm, NF = 6 dBm, f_c = 868 MHz; γ_{D} is chosen from one of the following values: $\gamma_{\text{D}} \in \{-6, -9, -12, -15, -17.5, -20\}$ (dBm), which absolutely depends on the utilized SF, e.g., if SF $= 7$ then $\gamma_{\text{D}} = -6$ dBm and SF $= 12$ then $\gamma_{\text{D}} = -20$ dBm, respectively. The range of the output, P^*_{tx}, in the first and second phases, on the other hand, are different. Particularly, the range of the optimal transmit power in the first training set denoted as $P^{*,1}_{\text{tx}}$ is from -100 dBm to 60 dBm while the range of P^*_{tx} in the second training set denoted as $P^{*,2}_{\text{tx}}$, is only from -40 dBm to 50 dBm. The chef reason behinds this difference is that $P^{*,2}_{\text{tx}}$ is obtained via Monte Carlo simulation which absolutely requires huge resources and efforts, thus, a limited search space is considered. $P^{*,1}_{\text{tx}}$, contrarily, is obtained by maths which is effortless compared to Monte Carlo method thus the range can be arbitrarily meaningful numbers.

In addition, in the second training set, $\Psi^{(2)}$, we have the average number of offspring per cluster, $\overline{C}_u = \{2, 6\}$, $u \in \{\text{Tho, Max}\}$, the radius of each cluster in Matérn cluster process, $r_{\text{Mat}} = \{2, 6\}$ m, and the standard derivation in Thomas cluster process, $\sigma^2_{\text{Tho}} = \{0.5, 3\}$. Moreover, also in the second training set, $\Psi^{(2)}$, both the static and idle power are impaired by an additive noise which follows either Gaussian distribution with zero mean and unit variance (in dBm) or

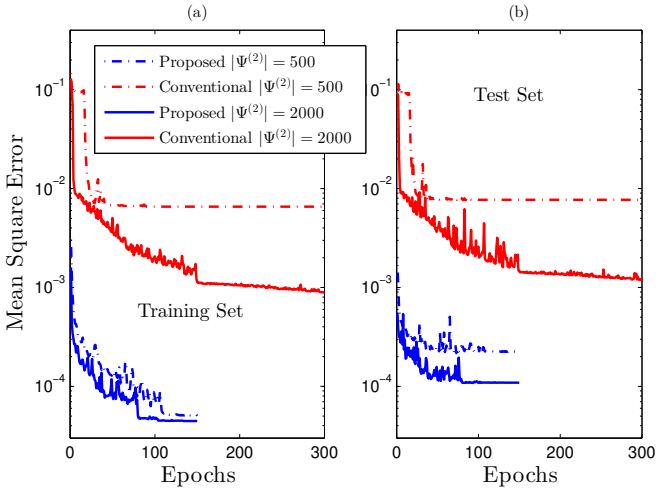

Fig. 3. Mean square error (MSE) versus number of epochs of training set (a) and test set (b) in the second training phase; the blue curves are proposed approach where the ANNs is pre-trained by model-based data, $\Psi^{(1)}$, and re-trained with practical data, $\Psi^{(2)}$; the red curves are the conventional approach which directly trains with $\Psi^{(2)}$. The ANNs architecture is $(\mathcal{L} = 4, \mathcal{N} = 55)$; $|\Xi| = 500$; Matérn cluster process with $\overline{C}_{\text{Mat}} = 2, r_{\text{Mat}} = 2$ and $\omega_x, x \in \{\text{sta,idl}\}$, follows Gaussian distribution.

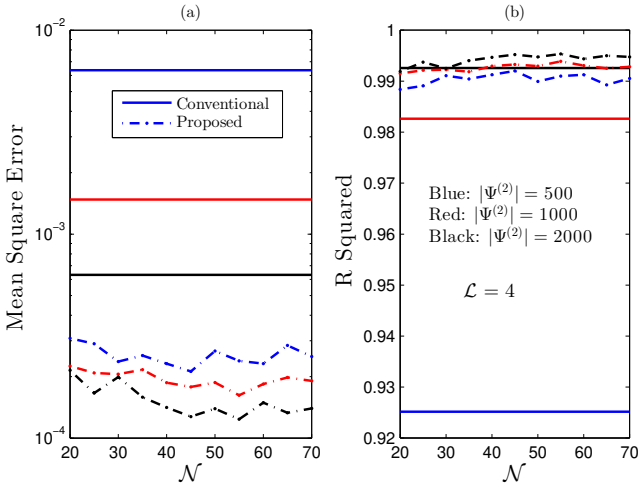

Fig. 4. MSE (a) and $\mathcal{R}^2$ (b) vs. number of neurons of each hidden layer, $\mathcal{N}$, of the test set with different training data size, i.e., $\left|\Psi^{(2)}\right|$; each point are chosen from the best epoch of that set up. Solid curves are conventional approach and dash dot curves are the proposed approach. Thomas cluster process with $\overline{C}_{\text{Tho}} = 2, \sigma_{\text{Tho}} = 0.5$ and the ideal static and idle power, 4 hidden layers, i.e., $\mathcal{L} = 4$, and $|\Xi| = 500$.

uniform distribution from [-0.5, 0.5] (dBm). The size of the 1st training set is equal to 30000, i.e., $|\Psi^{(1)}| = 30000$, while the size of the 2nd training set is less than 2000, i.e., $|\Psi^{(2)}| \leq 2000$, and the size of the test set denoted as $|\Xi|$ is always fixed at 500, i.e., $|\Xi| = 500$. Furthermore, the test set is solely available for the second training phase. It means that we do not evaluate the performance of the ANNs after the first training. It should be noted that the training set in both phases are able to cover most of the practical environment, i.e., urban or rural area, as well as the different applications/end-devices which are applied into various domains, i.e., smart home, smart city and smart health, etc [17]. In this section, without explicit stated, the performance from the best epoch is chosen and the size of the training data indicates the second training phase, $|\Psi^{(2)}|$.

A. Direct training versus double training approach

In this section, the performance between the conventional approach (direct training with the second data set, $\Psi^{(2)}$) and the proposed approach are examined.

Fig. 3 illustrates the mean square error of both the training and test set in second training phase versus number of epochs of the proposed approach and the conventional approach. To be more specific, the curves from the double training approach are firstly trained by 30000 model-based data, i.e., $\left|\Psi^{(1)}\right| = 30000$, in 150 epochs then re-training by either $\left|\Psi^{(2)}\right| = 500$ or $\left|\Psi^{(2)}\right| = 2000$, in another 150 epochs. The conventional approach, on the contrary, is direct training with the same amount of data of $\Psi^{(2)}$ in 300 epochs. We observe that our approach outperforms the conventional one in both training set and test set. Particularly, our approach is better almost ten times compared to its counterpart. In addition, it is expected that increasing training set slightly ameliorates the MSE performance under the proposed method as its weights and bias have already been configured to the near optimal values after the 1st training phase. The conventional one, differently, improves dramatically by soaring number of training data. Fig. 3 also reveals that increasing the number of epochs is not a wise choice to enhance the performance of the MSE especially in direct training approach where the MSE start constant from around 50 epochs (case $\left|\Psi^{(2)}\right| = 500$). Moreover, we experience that 300 and 150 epochs are sufficient to obtain the stable results for both the conventional and proposed approach. As a result, in the sequel, this number of epochs are yielded if no explicit stated.

Fig. 4 shows the performance of the MSE and $\mathcal{R}^2$ of both the conventional and the proposed methods versus the number of neurons, $\mathcal{N}$, with different values of $|\Psi^{(2)}|$. Each point of all curves are chosen from the best epoch that provides the minimum MSE or the maximum $\mathcal{R}^2$ among all epochs. The results are measured on the test set and it should be clarified that the best epoch over the test set is not necessary the same over the training set. We see that the performance of the conventional approach is constant with $\mathcal{N}$ and is regardless of the metrics, i.e., MSE and $\mathcal{R}^2$. It can be interpreted that the higher number of neurons, $\mathcal{N}$, is not necessary the better performance under the direct training approach. The double training approach, on the contrary, is fluctuated and the largest number or neurons, $\mathcal{N} = 70$, is not necessary the best performance, the best one, however, is case $\mathcal{N} = 55$ where the MSE = 1.236×10^{-4} and $\mathcal{R}^2 = 0.9953$ for case $\left|\Psi^{(2)}\right| = 2000$. It is interesting to point out that although increasing number of neurons does not always benefit, one is always true that the larger the training set the better the performance of the ANNs. Indeed, by rising $\left|\Psi^{(2)}\right|$ from 500 to 2000, the $\mathcal{R}^2$ of the conventional approach improves from 0.9252 to 0.9926 and is better than the $\mathcal{R}^2$ of proposed approach case $\left|\Psi^{(2)}\right| = 500$. Nevertheless, the MSE of the double training approach is never worse than the direct training one. Even taking the worst case from the proposed method into comparison, i.e., $\left|\Psi^{(2)}\right| = 500$, its performance is still better than the best one of the conventional approach, i.e., $\left|\Psi^{(2)}\right| = 2000$. In addition, in Fig. 4(b), we also experience that the $\mathcal{R}^2$ of the double training one is never lower than 0.985. It means that only

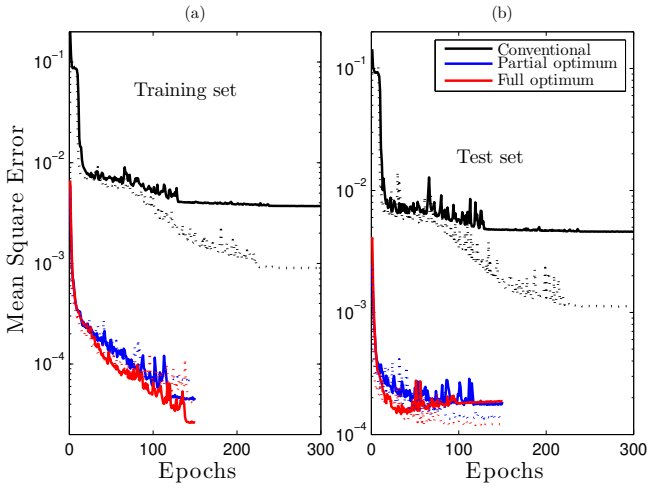

Fig. 5. MSE vs. number of epochs in training set (a) and test set (b) of $\Psi^{(2)}$; solid lines and dot lines are corresponding to $\left|\Psi^{(2)}\right| = 500$ and $\left|\Psi^{(2)}\right| = 1000$, respectively; $|\Xi| = 500$; the curves 'conventional' are direct training approach and selected from the best architecture based on $\Psi^{(2)}$; the curves 'partial optimum' and 'full optimum' are proposed approach and selected from the best architecture based on the $\Psi^{(1)}$ and $\Psi^{(1)} + \Psi^{(2)}$, respectively. Matérn cluster process with $\overline{C}_{\text{Mat}} = 6, r_{\text{Mat}} = 2$ and $\omega_x, x \in \{\text{sta,idl}\}$, follows uniform distribution.

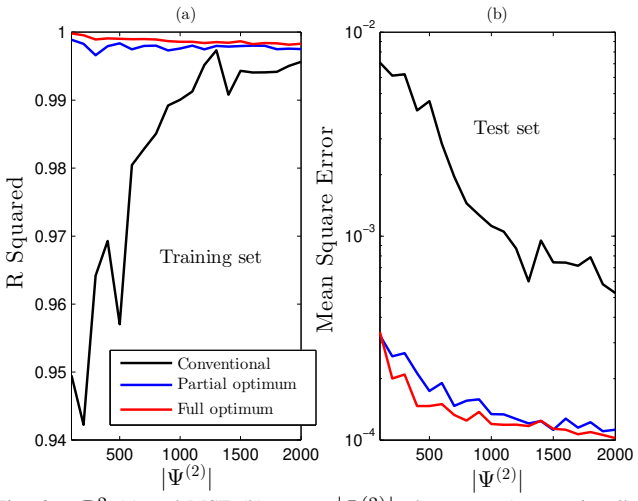

Fig. 6. $\mathcal{R}^2$ (a) and MSE (b) versus $\left|\Psi^{(2)}\right|$; the curves 'conventional' are direct training approach and selected from the best architecture based on the $\Psi^{(2)}$; the curves 'partial optimum' and 'full optimum' are proposed approach and selected from the best architecture based on the $\Psi^{(1)}$ and $\Psi^{(1)} + \Psi^{(2)}$, respectively. Thomas cluster process with average $\overline{C}_{\text{Tho}} = 6$ and $\sigma_{\text{Tho}} = 0.5$ and $\omega_x, x \in \{\text{sta,idl}\}$, follows uniform distribution.

around 1.5% the variability of the test set can not be explained by the ANNs.

B. Partial vs. Full optimum architecture

In the following, we are going to study the performance of the proposed approach by measuring the performance of the partial optimum, N_1^*, versus full optimum architecture, N_{1+2}^*. The partial optimum architecture means that we select the best topology based on $\Psi^{(1)}$, and solely utilize this architecture for $\Psi^{(2)}$. It is re-emphasized that the selection is based only on the training set. The full optimum architecture, N_{1+2}^*, selects the architecture that provides the best performance from both $\Psi^{(1)}$ and $\Psi^{(2)}$.

Fig. 5 unveils the performance of the MSE in both training set and test set of the second phase versus the number of epochs with various optimum architecture, i.e., partial and full optimum. In Fig. 5, in order to have a fair comparison, the curves from the conventional approach is also chosen from the best architecture in the same search space as the proposed approach, i.e., $\mathcal{T} = (\mathcal{L}_{\max} - \mathcal{L}_{\min})(\mathcal{N}_{\max} - \mathcal{N}_{\min})$. The conventional approach still underperforms both optimum architecture from the double training approach though the best architecture has been selected. In Fig. 5(a), it is crystal clear that the full optimum architecture is superior compared to the partial one. Nevertheless, the gap between these curves, in general, are negligible, for example, 3.99×10^{-5} vs. 6.79×10^{-5} for $\left|\Psi^{(2)}\right| = 2000$, and 1.34×10^{-4} vs. 1.19×10^{-4} for $|\Xi| = 500$. In addition, the expenses for this superior performance is quite expensive in terms of the consumed resources, namely, in order to attain the best architecture under full optimum approach, each topology requires to train exactly 300 epochs like the conventional approach while the partial optimum approach requires only half of each, i.e., 150 epochs. Mathematical speaking, assuming that the search space has $\mathcal{T} = (\mathcal{L}_{\max} - \mathcal{L}_{\min})(\mathcal{N}_{\max} - \mathcal{N}_{\min})$ architecture, then both the conventional and full optimum approach need $300\mathcal{T}$ epochs to attain the best architecture while the partial optimum approach, the epochs' requirements are only $150\,(\mathcal{T}+1)$. Again, we experience that surging the training data will monotonically increase the performance of the ANNs. As a consequence, in next figure, we investigate in detail the impact of training data on the performance of MSE and $\mathcal{R}^2$.

Fig. 6 illustrates the performance of $\mathcal{R}^2$ (a) and MSE (b) versus $|\Psi^{(2)}|$. We observe the same pattern as Fig. 4 that, in general, increasing training data will monotonically improve $\mathcal{R}^2$ and decline the MSE of all approaches, i.e., conventional, partial and full optimum. However, the pace of the improvement among these methods are different, namely, the direct training one enhances almost 10 times in terms of the MSE when training data rises from 100 to 2000 while for both partial and full optimum approach, it merely increases around 3 times. As for the $\mathcal{R}^2$, the performance of the partial and optimum approach are nearly stable and close to one while the remain one boosts around 5% in case the training data goes from 100 to 2000. From Fig. 6, it is apparent that in order to obtain the MSE equal to 10^{-4}, 2000 training data is sufficient for both optimum architecture. Furthermore, this figure also confirms the above statement that the gap between the partial and optimum approach is negligible.

Fig. 7 shows the optimal transmit power, P_{tx}^*, of three different ways versus the test set where the samples are sorted in ascending order. Looking at this figure, there is no surprising that the direct training approach experiences the biggest error between the observed output and the predicted one while the error under the partial and full optimum are relatively small. To better measure the error between the observation and the predicted one, let us utilize the mean absolute error (MAE) which refers to the arithmetic average of the absolute errors between the observed and predicted values. Then the MAE under the conventional, the partial and the full optimum approaches in Fig. 7 are 4.14, 1.47 and 1.35 dBm, respectively. It is noted that this MAE is computed based on the original range of the P_{tx}^*. Hence, if we compute the MAE after normalizing by (9), the MAE of these approaches are smaller,

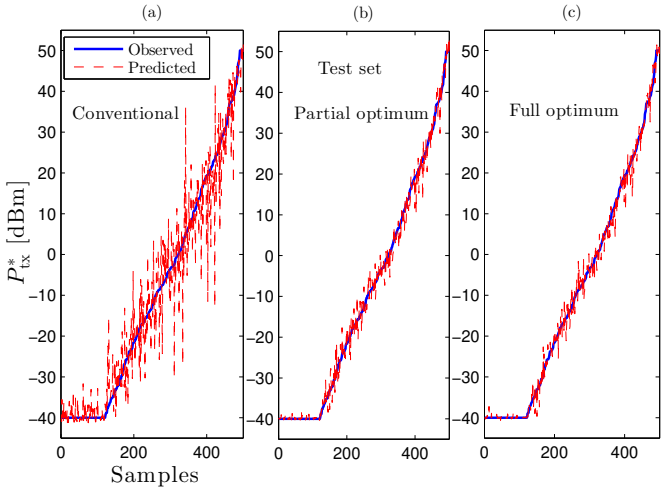

Fig. 7. P^*_{tx} versus the test set (sorted in ascending order) under various case studies, e.g., conventional (a), partial optimum (b) and fully optimum (c); the 'Observed' curves mean the observation output and the 'predicted' curves mean the predicted output from the ANNs. Matérn cluster process with $\overline{C}_{\text{Mat}} = 6, r_{\text{Mat}} = 2$ and $\omega_x, x \in \{\text{sta,idl}\}$, follows uniform distribution; $|\Psi^{(2)}| = 500$, and $|\Xi| = 500$.

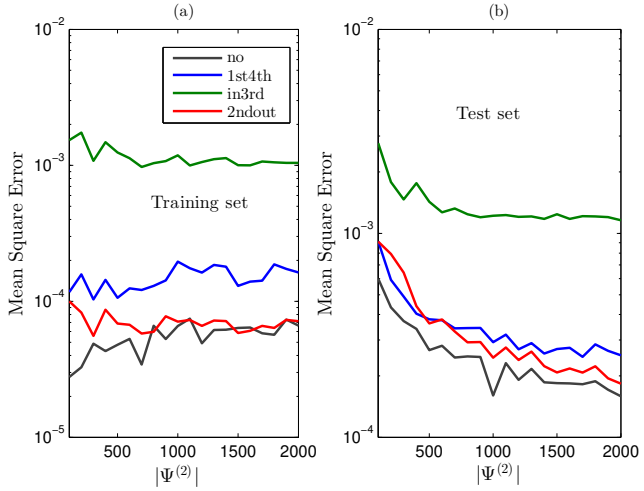

Fig. 8. MSE of the training set (a) and test set (b) versus training data size; the ANNs architecture is $(\mathcal{L} = 4, \mathcal{N} = 55)$ (partial optimum architecture); the 'no' curves denote without freezing any layers; the '1st4th' curves denote freezing from the 1st to the 4th hidden layers (all hidden layers); the 'in3rd' curves mean freezing from the input layer to the 3th hidden layer; the '2ndout' curves mean freezing from the 2nd hidden layer to the output layer. Thomas cluster process with $\overline{C}_{\text{Tho}} = 6$ and $\sigma_{\text{Tho}} = 0.5$ and $\omega_x, x \in \{\text{sta,idl}\}$, follows Gaussian distribution.

i.e., 0.046, 0.009 and 0.008 for the conventional, partial and full optimum architecture.

C. Re-train whole vs. parts of the ANNs

Fig. 8 illustrates the MSE in both training set and test set versus $|\Psi^{(2)}|$ by freezing some layers of the considered ANNs during the second training phase. In particular, we consider four distinct scenarios; i) re-train whole ANNs or no freezing denoted as 'no'; ii) freezing all hidden layers denoted as '1st4th'; iii) freezing from the input layer to the 3rd hidden layer denoted as 'in3rd'; and iv) freezing from the 2nd hidden layer to the output layer denoted as '2ndout'. In the last 3 cases, the ANNs only re-trains 2 layers, e.g., the input and output layers for case '1st4th', the input and the 1st hidden layer for case '2ndout' and the last hidden layer (the 4th hidden layer) and the output layer for case 'in3rd'. From the figure, we observe a fairly large gap between case 'in3rd' and others, in fact, its performance is worse than almost 10 times in both training and test set. The second worst case is '1st4th' where the ANNs only updates the weights and bias of the input and output layers. The best scenario is, of course, the case without freezing any layers. However, the difference between the best and the second best case, i.e., case '2ndout', is significantly small, especially when the training size is adequately large, i.e., around 2000 data. From Fig. 8, we conclude that the impact of freezing the input layer and/or hidden layers close to input on the performance of the ANNs is more serious than ones close to the output layer. Moreover, by carefully freezing some layers, the performance of the ANNs slightly reduces compared with fully re-trained the networks, thus, it is feasible to freeze some layers in order to save time and resources when re-training the neural networks.

VI. Conclusion

In this paper, the maximization of the EE respect to the transmit power is studied by combining the advantages of both model-based in wireless networks and the data-driven of the deep learning technique. Our findings show that the proposed training approach outperforms the conventional one almost 10 times under some scenarios. Moreover, the re-training process can be done either in whole or parts of the ANNs. Finally, the application of the proposed approach is general to cover most of EDs distributions as well as to overcome the hardware impairment at EDs.

References

[1] C. V. N. Index, "Cisco visual networking index: Global mobile data traffic forecast update, 2017-2022 white paper," Cisco, USA, 2019.

[2] L.-T. Tu and M. Di Renzo, "On the Energy Efficiency of Heterogeneous Cellular Networks with Renewable Energy Sources – A Stochastic Geometry Framework", *IEEE Trans. Wireless Commun.*, Early Access.

[3] C. Goursaud and J. M. Gorce, "Dedicated networks for IoT: PHY/MAC state of theart and challenges", *EAI Trans. IoT*, vol. 1, no. 1, 2015.

[4] A. Zappone, M. Di Renzo, M. Debbah, T. T. Lam and X. Qian, "Model-Aided Wireless Artificial Intelligence: Embedding Expert Knowledge in Deep Neural Networks for Wireless System Optimization," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 60-69, Sept. 2019.

[5] A. Zappone, M. Di Renzo and M. Debbah, "Wireless Networks Design in the Era of Deep Learning: Model-Based, AI-Based, or Both?," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331-7376, Oct. 2019.

[6] L. Y. Pratt, "Discriminability-based transfer between neural networks," *Proc. NIPS Advances in Neural Infor. Proc. Sys. 5*, pp. 204–211.

[7] L. Beltramelli, A. Mahmood, P. Osterberg and M. Gidlund, "LoRa beyond ALOHA: An Investigation of Alternative Random Access Protocols," *IEEE Trans. Ind. Informat.*, Early Access.

[8] B. Su, Z. Qin and Q. Ni, "Energy Efficient Uplink Transmissions in LoRa Networks," *IEEE Trans Commun.*, Early Access.

[9] R. M. Sandoval, A. Garcia-Sanchez and J. Garcia-Haro, "Optimizing and Updating LoRa Communication Parameters: A Machine Learning Approach," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 3, pp. 884-895, Sept. 2019.

[10] J. Cho, D. Hwang and K. Kim, "Improving TDoA Based Positioning Accuracy Using Machine Learning in a LoRaWan Environment," *Proc. IEEE ICOIN 2019*, Kuala Lumpur, Malaysia, 2019, pp. 469-472.

[11] M. Haenggi, Stochastic Geometry for Wireless Networks. Cambridge: Cambridge University Press, 2012.

[12] O. Georgiou and U. Raza, "Low Power Wide Area Network Analysis: Can LoRa Scale?", *IEEE Wireless Commun. Lett.*, vol. 6, no. 2, pp. 162 - 165, April 2017.

[13] L.-T. Tu, A. Bradai and Y. Pousset, "A New Closed-Form Expression of the Coverage Probability for Different QoS in LoRa Networks," *Proc IEEE ICC 2020*, Dublin, Ireland, 2020.

[14] E. Talbi, "Optimization of deep neural networks: a survey and unified taxonomy," [Online]. Available: https://hal.inria.fr/hal-02570804v2/document

[15] V. Subramanian, “Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch,”*Packt Publishing*,2018.
[16] D. P. Kingma, J. Ba, “Adam: A Method for Stochastic Optimization,” [Online]. Available: https://arxiv.org/pdf/1412.6980.pdf
[17] IEEE 802.16p-11/0014, *IEEE 802.16p Machine to Machine (M2M) Evaluation Methodology Document (EMD)*. [Online]. Available: http://ieee802.org/16/m2m/index.html