Deep learning–based resource allocation for secure transmission in a non-orthogonal multiple access network

Miao Zhang¹, Yao Zhang², Qian Cen³ and Shixun Wu¹

Abstract
Machine learning techniques, especially deep learning algorithms, have been widely utilized to deal with different kinds of research problems in wireless communications. In this article, we investigate the secrecy rate maximization problem in a non-orthogonal multiple access network based on deep learning approach. In this non-orthogonal multiple access network, the base station intends to transmit two integrated information: a confidential information to user 1 (the strong user) and a broadcast information to user 1 and user 2. In addition, there exists an eavesdropper that intends to decode the confidential information due to the broadcast nature of radio waves. Hence, we formulate the optimization problem as a secrecy rate maximization problem. We first solve this problem by employing convex optimization technique, then we generate the training, validation, and test dataset. We propose a deep neural network–based approach to learn to optimize the resource allocations. The advantages of the proposed deep neural network are the capabilities to achieve low complexity and latency resource allocations. Simulation results are provided to show that the proposed deep neural network approach is capable of reaching near-optimal secrecy rate performance with significantly reduced computational time, when compared with the benchmark conventional approach.

Keywords
Deep learning, non-orthogonal multiple access, resource allocation, physical layer security

Date received: 30 November 2021; accepted: 5 May 2022

Handling Editor: Yanjiao Chen

Introduction
Recently, artificial intelligence (AI)–driven techniques for wireless communications have attracted increasing research attentions.¹ The beyond fifth generation (B5G) and sixth generation (6G) wireless networks will emerge AI-driven use cases, including automatic driving, Internet of Things (IoT), and tactile communications. Machine learning techniques are the specific methods to realize AI, which has the capability to learn from previous experience and make decisions to the environment.²,³ Machine learning technique is divided into various frameworks, including regression algorithms, deep learning, random forest, and so on. Deep learning or deep neural network (DNN) is one of the most widely utilized machine learning technique,
which was developed to mimic the functions and organ-
izations of human brains.4

Multiple access techniques have been utilized in commercial wireless communication systems for a long history, from the first generation (1G) to 5G. Most wireless systems employ conventional orthogonal multiple access (OMA), for example, frequency division multiple access and code division multiple access, which is capable of serving only one user in a single orthogo-
nal resource block.3 Ultra-high data rate and massive connections are the basic demands of 5G and 6G wireless networks, as well as IoT networks.5 Non-orthogonal multiple access (NOMA) is considered as a promising technique to meet these demands.7,8 NOMA exploits the successive interference cancelation (SIC) approach at the receivers to serve multiple users in the same wireless resource block.5,9

Security is a critical and important problem for wire-
less systems. Due to the broadcast characteristic of information carrier, wireless communications are more vulnerable to be attacked.10,11 Conventional security methods employ secret keys to encrypt and decrypt confidential messages, which rely on the ultra-high complexities of specific mathematical problems.11,12

The conventional encryption methods are proven to be safe; however, one issue may arise that there exist risks in the procedures of secret key distributions and exchanges, due to the fact that secret keys are transmitted in plain text.13,14 Physical layer security is a technique to address this issue, which is capable of introducing additional security for secret key distributions and exchanges. The concept of physical layer security was first proposed by Shannon,15 then this research was push forwarded by Wyner and Csiszar.16,17 Theses work demonstrated that if the signal to noise plus inter-
ference ratio (SINR) or the signal to noise ratio (SNR) of the legitimate channel is larger than that of the eaves-
dropper’s channel, the eavesdropper cannot decode anything.

In the literature, most resource allocation designs are developed based on conventional convex optimization approaches.14,18–22 These methods employ iterative algorithms to obtain either optimal or sub-optimal solutions, which are high computational complexity and consume much time for computing. Therefore, it is challenging to deploy conventional resource allocation approaches to 5G and 6G systems, since they may not be able to meet the ultra-low latency demands of future wireless networks. Recently, deep learning–based resource allocation designs have drawn increasing research attentions. Zhou et al.23 investigated the deep learning–based resource allocation design in a cognitive radio network. Luo et al.24 studies the power minimization problem based on deep learning approach in a NOMA system. Sun et al.25 propose a DNN approach for interference management over interference limited channels. There are a large amount of machine learning tech-
niques and compared with other techniques, deep learning has several superiorities: (1) deep learning can be treated as a universal function approximator, which already has been employed to deal with different resource allocation problems;6,23 (2) along with the mini-batch gradient descent algorithm, deep learning is capable of saving computational resources, since other machine learning techniques, like support vector machine, require to calculate all the data at the same time;26,27 (3) in comparison with conventional design, the deep learning–based design can provide the near-optimal results within very short time slots, when the DNN is well trained.23

Motivated by the aforementioned aspects, in this article, we consider the secure transmission in a NOMA network, which consists of one base station, two users and one eavesdropper. The base station intends to send two information: one confidential information to user 1 and one broadcast information to user 1 and 2. It is assumed that the eavesdropper intends to wiretap the confidential information. Our aim is to maximize the secrecy rate of user 1 under the constraints of quality of service (QoS) of user 1 and 2. We first formulate the secrecy rate maximization problem and solve it by a lin-
eral fractional programming technique. Then, we obtain the training dataset through repeatedly running the simulation of this approach and feeding the training data into our DNN framework to process the training stage. When the training is finished, the weights of this DNN is determined and it establishes a mathematical relationship between the inputs and the corresponding outputs. We also generate a test dataset to evaluate the performance of the trained DNN. In particular, by comparing the performance of both schemes, we show that the DNN-based algorithm has the capabilities to reduce the computational time and achieve near-optimal performances. Our main contributions are summarized as follows:

1. To the best of our knowledge, none of the exist-
ing work has considered employing a DNN to design resource allocations for secure transmis-
sion in a NOMA network.
2. In this article, according to the characteristics of the power allocation coefficients in NOMA net-
work, we employ the novel cross-entropy as the cost function. The existing work of deep learning–based resource allocation designs employ the mean square error as the cost func-
tion,6,23,24 and utilize a min–max function to incorporate the power constraints, which may introduce additional computational complexity.
To address this issue, we employ the cross-entropy as the cost function, which has the capability to strictly satisfy the power allocation constraints.

3. We first formulate the secrecy rate maximization problem and solve this problem through conventional approach, which is exploited to generate the training dataset. Then, we train the proposed DNN with this dataset. Once the DNN is trained, we compare the secrecy rate performance and the computational time with the conventional approach. Finally, simulation results are provided to demonstrate that the proposed DNN approach can achieve near-optimal secrecy rate performance with significantly reduced computational time when compared with the conventional approach.

The rest of the content in this work will be formed as following: We present the system model in the “System model” section, whereas the resource allocation problem is formulated to maximize the secrecy rate and it is solved through conventional approach in the “Convex optimization-based approach” section. “The proposed DNN approach” section demonstrates the DNN-based resource allocation design. “Simulation results” section provides numerical results to evaluate the effectiveness of our proposed DNN approach and the conclusions are provided in the final section.

Notations: The upper and lower case boldface letters are utilized for representing matrices and vectors, respectively. The operation $(\cdot)^T$ denotes the transpose operation. The circularly Notations: symmetric complex Gaussian (CSCG) distribution is represented by $CN(\mu, \sigma^2)$ with mean $\mu$ and variance $\sigma^2$. $f(a)$ denotes the first derivative of function $f$ at $a$, whereas $[a]^+$ represents the operation $\max\{a, 0\}$. The optimal value of variable $x$ is expressed as $x^*$.

System model

In this work, we investigate a NOMA wireless network, which is shown in Figure 1. It is consists of four terminals: one base station, two users, and one eavesdropper. All the terminals are equipped with only one antenna. The base station intends to send confidential information to user 1, at the same time sends the broadcast information to user 1 and user 2. Due to the broadcast nature of wireless carrier, we assume that the eavesdropper tries to decode the confidential information. The channel state information (CSI) between the base station and user 1, user 2, as well as the eavesdropper can be presented as $h_1$, $h_2$, and $h_e$, respectively. We assume that the base station has perfect knowledge of all the CSI. It is worthy to note that the base station is capable of estimating the perfect CSI of all the channels by local oscillator power leakage from the radio frequency (RF) front end of the users’ and eavesdropper’s; the details of this process can be found in the study by Mukherjee and Swindlehurst.28 Let $P_t$ denote the transmit power, and the received signal at user 1, user 2, and the eavesdropper are expressed, respectively, as

$$y_1 = \sqrt{\gamma_1 P_t h_1 x_1} + \sqrt{\gamma_2 P_t h_2 x_2} + n_1$$

$$y_2 = \sqrt{\gamma_1 P_t h_2 x_1} + \sqrt{\gamma_2 P_t h_2 x_2} + n_2$$

$$y_e = \sqrt{\gamma_1 P_t h_e x_1} + \sqrt{\gamma_2 P_t h_e x_2} + n_e$$

where $x_1(\mathbb{E}\{|x_1|^2\} = 1)$ and $x_2(\mathbb{E}\{|x_2|^2\} = 1)$ are the confidential and broadcast symbols sent from the base station, respectively. The parameters $\gamma_1$ and $\gamma_2$ represent the power allocation coefficient for user 1 and user 2, respectively. The noise at user 1, user 2, and the eavesdropper are denoted by $n_1(\mathbb{E}\{|n_1|^2\} = \sigma_1^2)$, $n_2(\mathbb{E}\{|n_2|^2\} = \sigma_2^2)$, and $n_e(\mathbb{E}\{|n_e|^2\} = \sigma_e^2)$, respectively. We assume that $|h_1|>|h_2|$. Based on the NOMA principle, user 1 performs the SIC process. Furthermore, we follow a similar consumption as in the study by Zhang et al.29 that the broadcast information is first decoded at the eavesdropper. The SNR of decoding the confidential information at user 1 and the eavesdropper can be given as

$$\Gamma_{c,1} = \frac{\gamma_1 P_t |h_1|^2}{\sigma_1^2}$$

and

$$\Gamma_{c,e} = \frac{\gamma_1 P_t |h_e|^2}{\sigma_e^2}$$
respectively. The achieved secrecy rate at user 1 is defined as

\[ R_s = \log_2(1 + \Gamma_{c,1}) - \log_2(1 + \Gamma_{c,e}) \]  

(6)  

The SINR of decoding the broadcast information at user 1 and user 2 are expressed, respectively, as

\[ \Gamma_{b,1} = \frac{\gamma_1 P_t |h_1|^2}{\gamma_1 P_t |h_1|^2 + \sigma_i^2} \]  

(7)

and

\[ \Gamma_{b,2} = \frac{\gamma_2 P_t |h_2|^2}{\gamma_1 P_t |h_2|^2 + \sigma_i^2} \]  

(8)

The capacity of the broadcast information at user 1 and user 2 are expressed as

\[ R_{b,1} = (\log_2 1 + \Gamma_{b,1}) \]  

(9)  

and

\[ R_{b,2} = (\log_2 1 + \Gamma_{b,2}) \]  

(10)

Convex optimization–based approach

In this section, we formulate the secrecy rate maximization problem and present the conventional approach to solve it. This optimization problem can be formulated as

\[
\begin{align*}
\max_{\gamma_1, \gamma_2} & \quad \log_2(1 + \Gamma_{c,1}) - \log_2(1 + \Gamma_{c,e}) \\
\text{s.t.} & \quad R_{b,1} \geq q, R_{b,2} \geq q \\
& \quad P_t \geq E_s \\
& \quad \gamma_1 + \gamma_2 \leq 1 \\
& \quad \gamma_1 \geq 0, \gamma_2 \geq 0
\end{align*}
\]

(11)

where \( q \) is the minimum capacity requirement of the broadcast information. The above is not convex due to the non-convex objective function. We first introduce a slack variable \( \alpha \) and reformulate it into an epigraph form as

\[
\begin{align*}
\max_{\gamma_1, \gamma_2, \alpha} & \quad \alpha \\
\text{s.t.} & \quad R_{b,1} \geq q, R_{b,2} \geq q \\
& \quad \gamma_1 + \gamma_2 \leq 1 \\
& \quad \gamma_1 \geq 0, \gamma_2 \geq 0 \\
& \quad \gamma_1 \geq 0, \gamma_2 \geq 0
\end{align*}
\]

(12)

The problem in equation (12) is still non-convex due to the fractional constraint. To address this issue, we employ the Charnes–Cooper transformation,\(^{30}\) which is utilized to recast linear fractional problems into tractable forms. The first step of Charnes–Cooper transformation is to introduce a slack variable; in our design, we employ \( \xi \) as the slack variable and define the following relations

\[
\gamma_1 = \frac{\tilde{\gamma}_1}{\xi}, \quad \gamma_2 = \frac{\tilde{\gamma}_2}{\xi}, \quad \alpha = \frac{\tilde{\alpha}}{\xi}
\]

(13)

Then, the above problem can be reformulated as

\[
\begin{align*}
\max_{\tilde{\gamma}_1, \tilde{\gamma}_2, \alpha} & \quad \tilde{\gamma}_1 P_t |h_1|^2 \\
\text{s.t.} & \quad \xi + \frac{\tilde{\gamma}_1 P_t |h_1|^2}{\sigma_i^2} \geq \tilde{\alpha} \\
& \quad \xi + \frac{\tilde{\gamma}_1 P_t |h_1|^2}{\sigma_e^2} \leq 1 \\
& \quad P_t (\tilde{\gamma}_1 + \tilde{\gamma}_2) |h_1|^2 + \xi \sigma_i^2 \geq 2^\alpha \left( \tilde{\gamma}_1 P_t |h_1|^2 + \xi \sigma_i^2 \right) \\
& \quad P_t (\tilde{\gamma}_1 + \tilde{\gamma}_2) |h_2|^2 + \xi \sigma_e^2 \geq 2^\alpha \left( \tilde{\gamma}_1 P_t |h_2|^2 + \xi \sigma_e^2 \right) \\
& \quad \tilde{\gamma}_1 \geq 0, \tilde{\gamma}_2 \geq 0
\end{align*}
\]

(14)

The proposed DNN approach

In this section, we present the proposed resource allocation design based on the DNN approach. We propose a full connected DNN to learn the relationship between the input channel parameters and the output power allocation coefficients to maximize the secrecy rate at user 1. The proposed DNN is able to work and be robust to real-time scenarios, the reasons are: (1) we notice that all resource allocation optimization problems are solved by using different kind of algorithms, the fundamental works has shown that NNs are universal function approximators\(^ {33}\) and have the remarkable capabilities of algorithmic learning;\(^ {34}\) (2) prior works demonstrated that DNN techniques have the capability to substantially reduce the computational complexity and processing time for a variety of problems in wireless communications, that is, resource allocation optimizations,\(^ {6,23,25}\) channel estimations,\(^ {35}\) and physical layer designs;\(^ {36}\) (3) the increased processing capability of computers due to recent advancements of processors and massively parallel processing architectures, which confirms the robustness of computers to process a huge amount of data within a very short time frame.\(^ {37}\)
Note that for other resource allocation design scenarios, it is difficult to apply the proposed approach directly, due to the fact that different system models and optimization problems have different inputs and outputs. It only needs to modify the structure of DNN and then it can be utilized for other scenarios. Furthermore, a DNN technique cannot handle the optimization problems in dynamic environments efficiently; a reasonable method may be a combination of several machine learning techniques including DNN, reinforcement learning, or other algorithms.

As shown in Figure 2, our proposed DNN has three parts: the input layer, multiple hidden layers, and the output layer. In this work, the absolute value of channel coefficients $|h_1|$, $|h_2|$, and $|h_e|$ are chosen to be the input parameters, whereas the power allocation coefficients $g_1$ and $g_2$ are assumed to be the outputs. The DNN can be seen as a function approximator to map the relations between the aforementioned inputs and outputs. First, we define the mapping of the maximization problem as

$$g_j / C_3 = f(h_1, h_2, h_e)$$

where $z^{(i+1)}$ represents the linear combination of the $(i+1)$th layer and $n^{(i+1)}$ is the activation value of the $(i+1)$th layer. $\Theta^{(i)}$ and $\beta^{(i)}$ represents the weights and bias at the $i$th layer. $a(\cdot)$ denotes the activation function; in this design, it is assumed that all the hidden layers utilize the rectified linear unit (ReLU) as the activation function, that is, $a(z) = \max\{z, 0\}$. Whereas we choose the softmax as the activation function for the output layer, which is written as

$$a(z) = \frac{e^{z_j}}{\sum_{j=1}^2 e^{z_j}}$$

Assume the proposed DNN has $I$ layers in total, then the mapping function of this DNN can be expressed as

$$y = f(S, \Theta, \beta)$$

where $S = [|h_1|, |h_2|, |h_e|]$, $\Theta = [\Theta^{(1)}, ..., \Theta^{(I-1)}]$, and $\beta = [\beta^{(1)}, ..., \beta^{(I-1)}]$. Our aim is to train the DNN to enable the mapping function in equation (18) to obtain similar outputs with that of equation (15). To achieve this target, we employ cross-entropy as the cost function, as the activation function of the output layer is set to softmax. The cost function can be expressed as

$$J(\Theta, \beta) = -\frac{1}{M} \sum_{m=1}^M \sum_{j=1}^2 \gamma_{j,m} \log(y_{j,m})$$

where $M$ is the number of training batch size, $\gamma_{j,m}$ is the optimal $g_j$, $j = 1, 2$ in the $m$th dataset, whereas $y_{j,m}$, $j = 1, 2$ is the $j$th DNN output obtained from the
Lemma 1. To minimize the cost function, the back propagation–based gradient descent algorithm can be employed, where the 1th layer weights $\Theta^{(i)}$ and bias $\beta^{(i)}$ can be iteratively updated by

$$
\Theta^{(i)} = \Theta^{(i)} - \frac{\epsilon}{M} \sum_{m=1}^{M} \left[ \Delta_{m}^{(i+1)} \right] \left( n_{m}^{(i)} \right)^{T} \tag{20}
$$

$$
\beta^{(i)} = \beta^{(i)} - \frac{\epsilon}{M} \sum_{m=1}^{M} \Delta_{m}^{(i+1)} \tag{21}
$$

where $\epsilon$ is the learning rate and $\Delta_{m}^{(i+1)}$ is the gradient of the cost function with respect to the $m$th input. To enable the DNN to achieve near-optimal solutions, the above cost function should be minimized.

Proof. Based on the chain rule and the cost function $J(\Theta, \beta)$, we derive the following formulations

$$
\frac{\partial J(\Theta, \beta)}{\partial \Theta^{(i)}} = \frac{\partial J(\Theta, \beta)}{\partial z^{(i+1)}} \frac{\partial z^{(i+1)}}{\partial \Theta^{(i)}} = \frac{1}{M} \sum_{m=1}^{M} \Delta_{m}^{(i+1)} \left( n_{m}^{(i)} \right)^{T} \tag{22}
$$

$$
\frac{\partial J(\Theta, \beta)}{\partial \beta^{(i)}} = \frac{\partial J(\Theta, \beta)}{\partial z^{(i+1)}} \frac{\partial z^{(i+1)}}{\partial \beta^{(i)}} = \frac{1}{M} \sum_{m=1}^{M} \Delta_{m}^{(i+1)} \tag{23}
$$

As it is easy to obtain $n_{m}^{(i)}$ from the feed-forward calculations, we can derive $\Delta_{m}^{(i)}$ based on the chain rule, as

$$
\Delta_{m}^{(i)} = \frac{\partial J(\Theta, \beta)}{\partial z_{m}^{(i)}} = \frac{\partial J(\Theta, \beta)}{\partial \Theta^{(i)}} \frac{\partial \Theta^{(i)}}{\partial z_{m}^{(i+1)}} \frac{\partial z_{m}^{(i+1)}}{\partial n_{m}^{(i)}} \frac{\partial n_{m}^{(i)}}{\partial \Delta_{m}^{(i+1)}} = \left[ \left( \Theta^{(i)} \right)^{T} \Delta_{m}^{(i+1)} \right] \cdot d \left( z_{m}^{(i)} \right) \tag{24}
$$

Based on the back propagation policy, the weights and bias of the DNN are updated one by one from the output layer to the input layer. Then, we consider the gradient descent algorithm and introduce the learning rate $\epsilon$ to the above equations. The weights and bias at the $i$th layer are updated by the following equations

$$
\Theta^{(i)} = \Theta^{(i)} - \frac{\epsilon}{M} \sum_{m=1}^{M} \left[ \Delta_{m}^{(i+1)} \right] \left( n_{m}^{(i)} \right)^{T} \tag{25}
$$

$$
\beta^{(i)} = \beta^{(i)} - \frac{\epsilon}{M} \sum_{m=1}^{M} \Delta_{m}^{(i+1)} \tag{26}
$$

which completes the proof of Lemma 1.

The implementation of the proposed DNN approach are divided into three parts: preparing, training, and evaluating. The detailed information of these parts are summarized in Algorithm 1.

**Algorithm 1.** The DNN-based resource allocation design.

Preparation:
1. Generate the training, validation, and test datasets through repeatedly run the simulation of the conventional design;

Training:
1. Initialize, randomly generate the initial values of weights $\Theta$ and the bias $\beta$, choose the number of learning rate $\alpha$ and the size of each mini-batch $M$;
2. For each epoch:
3. For each mini-batch: Feed the whole batch data for training, where $S = [S_{1}, ..., S_{M}]$ is the input data, whereas $y_{1}^{*} = [y_{1}^{*}, ..., y_{M}^{*}]$ and $y_{2}^{*} = [y_{2}^{*}, ..., y_{2}^{*}]$ are the output training data;
4. Update the weights $\Theta^{(i)}$ and the bias of the $i$th layer through equations (20) and (21), respectively;
5. End for;
6. End for;
7. When training is completed, save the DNN model.

Evaluating:
1. Call the DNN and input the channel coefficients $S_{test}$ from the test dataset to obtain the output power allocations;
We randomly generate the initial parameters of the DNN by Gaussian distribution with zero mean and 0.5 variance and we employ the Adam optimizer for the training process. The learning rate is assumed to be $E = 10^{-4}$. All the calculations are performed through the central processing unit (CPU) on the same computer with Intel Core i9-12900 K CPU and 64 GB random access memory.

Figure 3 depicts the cross-entropy obtained by training and validation datasets versus the number of training epochs. Both the curves first decrease as the number of epochs increases and then stay constant. Note that the validation dataset is utilized to calculate the cross-entropy only, it does not contribute to the training. The purpose of introducing the validation dataset in the training process is to observe whether over-fitting is occurred. When the DNN learns the training dataset very precisely, it may learn some noise as well. If we introduce additional dataset, the DNN may not be able to fit the additional data. As seen in this figure, it is obvious that over-fitting does not happen. It is due to the fact that over-fitting happens when a machine learning model is more accurate in fitting a particular set of data but less accurate in additional data. The validation data do not participate in training process and can be seen as an additional data. Over-fitting occurs if the cross-entropy of the validation data increases while that of the training data steadily decreases. Therefore, the cross-entropy of the validation data remains constant confirming that over-fitting does not occur.

Next, Figure 4 presents the achieved secrecy rate versus transmit power obtained by conventional and DNN approaches. The QoS of the broadcast information is set to 5 bps/Hz. As shown in this figure, the achieved secrecy rates rise as the transmit power increases. In addition, the proposed DNN approach is capable of achieving a similar secrecy rate performance as the optimal solution. However, there is a performance gap between the optimal solution and our proposed DNN approach, which is because the training errors always existed.

Then, the achieved secrecy rate versus the QoS of broadcast information obtained by conventional and DNN approaches were provided in Figure 5. The
Transmit power is set to 30 dBm. As seen in this figure, the achieved secrecy rate declines as the capacity requirements of broadcast information increases. Similar to the previous figure, it is shown in this figure that the proposed DNN approach has a near-optimal performance and there also exists a performance gap between the conventional and DNN approach.

Next, Table 1 demonstrates the achieved secrecy rates of both approaches with different transmit power, whereas Table 2 presents the computational time of both approaches with different transmit power. These tables show a more detailed information of Figure 4. Note that the secrecy rate is obtained through averaging all 3000 results of the test dataset, whereas the computational time is the summation of 3000 results. In these tables, “Transmit power” means the maximum available transmit power at the base station, “DNN” stands for the results of our proposed DNN approach, “Conventional” means the results of conventional optimization approach, and “Ratio” is the ratio of “DNN” divided by “Conventional.” From the tables, it is obvious that our proposed DNN achieves no less than 95% of the optimal performance, while requiring no more than 1% of the computational time than the conventional optimization approach.

Next, Table 3 depicts the achieved secrecy rates of both approaches with different QoS demands of broadcast information, whereas Table 4 illustrates the computational time of both approaches with QoS demands of broadcast information. These tables show a more detailed information of Figure 5. The calculation methods of secrecy rate and computational time, as well as the definition of “DNN,” “Conventional,” and “Ratio” are same as those of Tables 1 and 2. In these tables, “QoS demands” denote the capacity requirements of broadcast information. As shown in these tables, the proposed DNN approach can achieve more than 97% secrecy rate performance with no more than 1% computational time in comparison with the conventional approach.

Then, Figure 6 presents the achieved secrecy rate against different number of hidden layers (left axis, red curve), as well as the computational time of the test dataset versus different number of hidden layers (right axis, pink curve). As seen in this figure, we can conclude that increasing the number of hidden layers affects only few of the secrecy rate performance, where the differences are within a range of 1%. This confirms that introducing number of hidden layers cannot bring much performance improvement. In addition, the achieved secrecy rates go up and down as the number of hidden layers rises; since the initial parameters of the DNN as well as the optimizer are randomly generated, different combinations may result in different
performances. However, more hidden layer will introduce more computational time, as the presented by the pink curve. This is due to the fact that the computational complexity increases as the number of hidden layers rises.

Finally, Figure 7 demonstrates the probabilities of the QoS constraints not satisfied in the test dataset versus different transmit power, whereas Figure 8 presents the probabilities of the QoS constraints not satisfied in the test dataset versus different capacity requirements. The results of these two figures are calculated by counting the total number of the non-satisfied constraints in the test dataset, then divided by the total number of QoS constraints of the same dataset. The simulation parameters of Figure 7 is the same as that of Figure 4, and the assumptions of Figure 8 is the same as Figure 5. As shown in Figure 7, the probability of the non-satisfied constraints is decreasing as the transmit power increases. In addition, from Figure 8, we can see this probability increases as the capacity requirement rises. Overall, for the aspects of constraints satisfaction, we can conclude that the proposed DNN approach is capable of satisfying more than 93% of the constraints. In particular, the DNN is a function approximator that can automatically find the relation between the input and output. During training process, the value of cost function is not able to reach zero. In other words, the training errors cannot be completely eliminated. Hence, the gap between the optimal solution and the output of the DNN may exist, because of which the QoS constraints cannot be strictly satisfied.

Conclusion

In this article, we proposed a DNN-based resource allocation design to maximize the secrecy rate in a NOMA network. We first formulated the problem and solved it through conventional convex optimization approach. Then, we developed the DNN-based approach, where the cross-entropy cost function is utilized to incorporate the power allocation constraint without additional operations. Simulation results were provided to evaluate the performance of our proposed DNN approach. We demonstrated that the proposed approach has the capability to achieve no less than 95% secrecy rate performance with no more than 1% computational time in comparing with the benchmark convex optimization approach.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the National...
Nature Science Foundation of China under Grant no. 62101080, in part by the Science and Technology Research Program of Chongqing Municipal Education Commission of China under Grant nos KJQN202100738 and KJQN202000703, in part by the Natural Science Foundation of Chongqing under Grant no. cstc2021jcjy-msxmX0017, and in part by the Research Start Up Funding of Chongqing Jiaotong University under Grant nos 2020020070 and XJ2021000501.

**ORCID iD**

Miao Zhang  https://orcid.org/0000-0001-9959-7238

**References**

1.  Letaief KB, Chen W, Shi Y, et al. The roadmap to 6G: AI empowered wireless networks. *IEEE Commun Mag* 2019; 57(8): 84–90.
2.  Goodfellow I, Bengio Y, Courville A, et al. *Deep learning*. Vol. 1. Cambridge: MIT Press, 2016.
3.  Bishop CM. *Pattern recognition and machine learning*. New York: Springer, 2006.
4.  Chen M, Challita U, Saad W, et al. Machine learning for wireless networks with artificial intelligence: a tutorial on neural networks. *IEEE Commun Surv Tutor* 2019; 21(4): 3039–3071.
5.  Ding Z, Lei X, Karagiannidis GK, et al. A survey on non-orthogonal multiple access for 5G networks: research challenges and future trends. *IEEE J Select Areas Commun* 2017; 35(10): 2181–2195.
6.  Zhang M, Cumanan K, Thiylagalingam J, et al. Exploiting deep learning for secure transmission in an underlay cognitive radio network. *IEEE Trans Veh Tech* 2021; 70(1): 726–741.
7.  Li X, Wang Q, Liu M, et al. Cooperative wireless-powered NOMA relaying for B5G IoT networks with hardware impairments and channel estimation errors. *IEEE Internet of Things J* 2021; 8(7): 5453–5467.
8.  Li X, Zheng Y, Alshehri MD, et al. Cognitive AmBC-NOMA IoV-MTS networks with IQE: reliability and security analysis. *IEEE Trans Intell Trans Sys* 2021; 2019: 1–12.
9.  Li X, Zhao M, Zeng M, et al. Hardware impaired ambient backscatter NOMA systems: reliability and security. *IEEE Trans Commun* 2021; 69(4): 2723–2736.
10. Zhang M, Cumanan K and Burr AG. Secure energy efficiency optimization for MISO cognitive radio network with energy harvesting. In: *9th international conference on wireless communications and signal processing (WCSP)*. Nanjing, China, 11–13 October 2017. New York: IEEE.
11. Chu Z, Zhou F, Xiao P, et al. Resource allocation for secure wireless powered integrated multicast and unicast services with full duplex self-energy recycling. *IEEE Trans Wireless Commun* 2019; 18(1): 620–636.
12. Chu Z, Zhu Z, Johnston M, et al. Simultaneous wireless information power transfer for MISO secrecy channel. *IEEE Trans Veh Technol* 2016; 65(9): 6913–6925.
13. Chu Z, Nguyen HX, Le TA, et al. Secure wireless powered and cooperative jamming D2D communications. *IEEE Trans Green Commun Netw* 2017; 2(1): 1–13.
14. Zhang M, Cumanan K, Thiylagalingam J, et al. Energy efficiency optimization for secure transmission in MISO cognitive radio network with energy harvesting. *IEEE Access* 2019; 7: 126234–126252.
15. Shannon C. Communication theory of secrecy systems. *Bell Syst Tech J* 1949; 28(4): 656–715.
16. Wyner AD. The wire-tap channel. *Bell Syst Tech J* 1975; 54(8): 1355–1387.
17. Csiszár I and Korner J. Broadcast channels with confidential messages. *IEEE Trans Inf Theory* 1978; 24(3): 339–348.
18. Chu Z, Xing H, Johnston M, et al. Secrecy rate optimizations for a MISO secrecy channel with multiple multi-antenna eavesdroppers. *IEEE Trans Wireless Commun* 2016; 15(1): 283–297.
19. Chu Z, Zhu Z, Zhou F, et al. Intelligent reflecting surface assisted wireless powered sensor networks for internet of things. *IEEE Trans Commun* 2021; 69(7): 4877–4889.
20. Niu H, Guo D, Huang Y, et al. Robust energy efficiency optimization for secure MIMO SWIPT systems with non-linear EH model. *IEEE Commun Lett* 2017; 21(12): 2610–2613.
21. Li Y, Zhao R, Wang Y, et al. Artificial noise aided precoding with imperfect CSI in full-duplex relaying secure communications. *IEEE Access* 2018; 6: 44107–44119.
22. Li Y, Zhao R, Tan X, et al. Secrecy performance analysis of artificial noise aided precoding in full-duplex relay systems. In: *GLOBECOM 2017 - 2017 IEEE global communications conference*, Singapore, 4–8 December 2017, pp. 1–6. New York: IEEE.
23. Zhou F, Zhang X, Hu RQ, et al. Resource allocation based on deep neural networks for cognitive radio networks. In: *IEEE/CIC international conference on communications in China (ICCC)*, Beijing, China, 16–18 August 2018, pp. 40–45. New York: IEEE.
24. Luo J, Tang J, So DK, et al. A deep learning-based approach to power minimization in multi-carrier NOMA with SWIPT. *IEEE Access* 2019; 7: 17450–17460.
25. Sun H, Chen X, Shi Q, et al. Learning to optimize: training deep neural networks for interference management. *IEEE Trans Signal Process* 2018; 66(20): 5438–5453.
26. Domingos PM. A few useful things to know about machine learning. *ACM Commun* 2012; 55(10): 78–87.
27. Zhang C, Patras P and Haddadi H. Deep learning in mobile and wireless networking: a survey. *IEEE Commun Surv Tutor* 2019; 21(3): 2224–2287.
28. Mukherjee A and Swindlehurst AL. Detecting passive eavesdroppers in the MIMO wiretap channel. In: *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, Kyoto, Japan, 25–30 March 2012, pp. 2809–2812. New York: IEEE.
29. Zhang Y, Wang HM, Yang Q, et al. Secrecy sum rate maximization in non-orthogonal multiple access. *IEEE Commun Lett* 2016; 20(5): 930–933.
30. Charnes A and Cooper WW. Programming with linear fractional functionals. *Naval Res Logist Quart* 1962; 9(3–4): 181–186.
31. Pei Y, Liang YC, Zhang L, et al. Secure communication over MISO cognitive radio channels. *IEEE Trans Wireless Commun* 2010; 9(4): 1494–1502.
32. Boyd S and Vandenberghe L. *Convex optimization*. Cambridge: Cambridge University Press, 2004.
33. Hornik K, Stinchcombe M, White H, et al. Multilayer feedforward networks are universal approximators. *Neural Networks* 1989; 2(5): 359–366.
34. Reed S and De Freitas N. Neural programmer-interpreters, 2015, https://arxiv.org/abs/1511.06279
35. Ye H, Li GY and Juang BH. Power of deep learning for channel estimation and signal detection in OFDM systems. *IEEE Wireless Commun Lett* 2018; 7(1): 114–117.
36. O’Shea T and Hoydis J. An introduction to deep learning for the physical layer. *IEEE Trans Cognit Commun Netw* 2017; 3(4): 563–575.
37. Jiang C, Zhang H, Ren Y, et al. Machine learning paradigms for next-generation wireless networks. *IEEE Wireless Commun* 2017; 24(2): 98–105.
38. Kingma DP and Ba J. Adam: a method for stochastic optimization, 2014, https://arxiv.org/pdf/1412.6980.pdf
39. Chicco D. Ten quick tips for machine learning in computational biology. *Biodata Mining* 2017; 10(1): 35.
40. Leinweber DJ. Stupid data miner tricks: overfitting the s&p 500. *J Invest* 2007; 16(1): 15–22.