

Deep Neural Network Based Precoding for Wiretap Channels With Finite Alphabet Inputs

Mücahit Gümüş¹, *Student Member, IEEE*, and Tolga M. Duman², *Fellow, IEEE*

Abstract—We consider secure transmission over multi-input multi-output multi-antenna eavesdropper (MIMOME) wiretap channels with finite alphabet inputs. We use a linear precoder to maximize the secrecy rate, which benefits from the generalized singular value decomposition to obtain independent streams and exploits the function approximation abilities of deep neural networks (DNNs) for solving the required power allocation problem. It is demonstrated that the DNN learns the optimal power allocation without any performance degradation compared to the conventional technique with a significant reduction in complexity.

Index Terms—Deep neural networks, physical layer security, MIMOME wiretap channels.

I. INTRODUCTION

WIRELESS communication systems have inherent security problems due to the broadcast nature of the wireless medium. A suitable model to address such security issues at the physical layer is the wiretap channel introduced in [1], which consists of a transmitter “Alice”, a legitimate receiver “Bob” and an eavesdropper “Eve”. In [1], the secrecy capacity, which is the maximum rate at which data can be securely transmitted between Alice and Bob while keeping Eve ignorant, is computed when Eve’s channel is a degraded version of the legitimate channel. In [2], this result is generalized to the case of non-degraded channels, and existence of channel codes that simultaneously guarantee robustness against errors and a certain degree of confidentiality is shown. More recently, the physical layer security problem is considered for multi-input multi-output (MIMO) systems with Gaussian inputs and precoding based on generalized singular value decomposition (GSVD) is performed [3]. In addition to the use of the multiple antennas, injection of artificial noise is used as another attractive technique to achieve secrecy [4].

In [3], it is proven that secrecy capacity can be achieved with Gaussian inputs, however, in practical scenarios, finite alphabet signals are used. With this motivation, precoding for finite alphabet inputs is considered in [5]–[7]. The joint design

of artificial noise and precoder is employed in [5]. The channel decomposition technique of [3] is adopted to the case of finite alphabet signals and an iterative algorithm based on dividing the main optimization into subproblems and connecting them through the minimization of a latent variable is proposed in [6].

Deep neural networks (DNNs) have recently attracted great interest in various areas. Their unbeatable performance in classification problems and excellent function approximation abilities widen the utilization of different deep learning techniques. As such, there have been recent studies on communication systems which employ DNNs in many tasks including precoding, decoding of linear codes and end-to-end communication system design [8]–[10]. In [11], various signal processing blocks at the transmitter and the receiver are replaced with a single DNN based autoencoder, and an overall end-to-end communication system is obtained. Furthermore, the solution is implemented with software defined radios and tested over the air.

Deep learning has been used in physical layer security problems as well [12]–[14]. Through the utilization of the DNNs in all of the parties in a wiretap channel, and exploiting suitable cost functions, constellation clusters that confuse Eve have been learned at the transmitter in [12]. Deep learning is used with quality of service (QoS) functions to increase the secrecy rates in [13].

In this letter, we aim at designing a low complexity precoder based on the use of DNNs to maximize the secrecy rate of multi-input multi-output multi-antenna eavesdropper (MIMOME) Gaussian wiretap channels with finite alphabet inputs. Similar to the application in [6], we decompose channel into independent streams through singular vectors. And then, we employ a specifically designed fully-connected DNN using the generalized singular values and the available power level to optimize the power allocation strategy as an alternative to the iterative algorithm which has a very high complexity. We show that the proposed DNN with a small number of nodes and layers learns the optimal power allocation without any performance degradation compared to the iterative solution while reducing the amount of computations to a suitable level for real-time implementations.

The remainder of the letter is organized as follows. We describe the system model and formulate the problem in Section II. We give a summary of GSVD and conventional iterative power allocation scheme in Section III. In Section IV, we explain the proposed DNN based precoder and conduct a complexity analysis. Section V provides a DNN example and numerical results revealing the performance of the proposed algorithms. Finally, we conclude the letter in Section VI.

Manuscript received February 27, 2021; accepted April 20, 2021. Date of publication April 28, 2021; date of current version August 9, 2021. The associate editor coordinating the review of this article and approving it for publication was N. Zhao. (*Corresponding author: Mücahit Gümüş.*)

Mücahit Gümüş is with the Communication and Information Technologies Division, ASELSAN Inc., 06800 Ankara, Turkey, and also with the Department of Electrical and Electronics Engineering, Bilkent University, TR-06800 Ankara, Turkey (e-mail: mgumus@ee.bilkent.edu.tr).

Tolga M. Duman is with the Department of Electrical and Electronics Engineering, Bilkent University, TR-06800 Ankara, Turkey (e-mail: duman@ee.bilkent.edu.tr).

Digital Object Identifier 10.1109/LWC.2021.3076398

Throughout the letter, vectors are indicated with bold lowercase letters, and matrices shown by bold uppercase letters. The notations, $\text{tr}(\cdot)$, $(\cdot)^H$, $\|\cdot\|$, and \mathbb{E} are used for trace, Hermitian, L_2 norm, and expectation operations, respectively. The operation $(\cdot)^+$ stands for $\max(0, x)$.

II. SYSTEM MODEL AND PROBLEM STATEMENT

We consider MIMOME wiretap channel for which Alice, Bob and Eve have m_a , m_b and m_e antennas, respectively. The received signals of Bob and Eve are expressed as

$$\begin{aligned} \mathbf{y}_b &= \mathbf{H}_b \mathbf{x} + \mathbf{n}_b, \\ \mathbf{y}_e &= \mathbf{H}_e \mathbf{x} + \mathbf{n}_e, \end{aligned} \quad (1)$$

where $\mathbf{H}_b \in \mathbb{C}^{m_b \times m_a}$ denotes the channel matrix between Alice and Bob, and $\mathbf{H}_e \in \mathbb{C}^{m_e \times m_a}$ represents the channel between Alice and Eve. The elements of channel matrices are modeled as zero mean and unit variance circularly symmetric complex Gaussian random variables (i.e., Rayleigh fading), and both \mathbf{H}_b and \mathbf{H}_e are assumed to be known at the transmitter and their respective receivers. We note that the assumption on the availability of both Bob's and Eve's CSI at Alice is valid in many communication scenarios, for example, when both the legitimate receiver and the eavesdropper are participants of a wireless network (as in cellular systems), and it is adopted in many studies on physical layer security, e.g., [6], [15]. \mathbf{n}_b and \mathbf{n}_e denote additive circularly symmetric complex white Gaussian noise terms with zero means and variances σ_b^2 and σ_e^2 , respectively. $\mathbf{x} \in \mathbb{C}^{m_a \times 1}$ is the transmitted signal given by

$$\mathbf{x} = \mathbf{W} \mathbf{s}, \quad (2)$$

where $\mathbf{W} \in \mathbb{C}^{m_a \times m_a}$ is the precoder matrix and $\mathbf{s} \in \mathbb{C}^{m_a \times 1}$ is the vector of data symbols with zero mean and unit variance, picked from a finite cardinality input. The input power constraint is given by $\text{tr}(\mathbb{E}[\mathbf{x}\mathbf{x}^H]) \leq P_T$. The secrecy rate is calculated by using [7]

$$R_{\text{sec}}(\mathbf{W}) = (I(\mathbf{x}; \mathbf{y}_b) - I(\mathbf{x}; \mathbf{y}_e))^+, \quad (3)$$

where $I(\mathbf{x}; \mathbf{y})$ denotes the mutual information between \mathbf{x} and \mathbf{y} (computed under the assumption that the CSI is available). The objective is to solve for \mathbf{W} that maximizes the secrecy rate $R_{\text{sec}}(\mathbf{W})$ under the average power constraint $\text{tr}(\mathbb{E}[\mathbf{x}\mathbf{x}^H]) \leq P_T$.

III. LINEAR PRECODING BASED ON GENERALIZED SINGULAR VALUE DECOMPOSITION

The maximization problem given in Section II is non-convex, making it difficult to solve. Therefore, a linear precoding scheme based on GSVD is proposed in [6] in which the GSVD expression of [3] is used, i.e.,

$$\begin{aligned} \mathbf{H}_b &= \Psi_b \Sigma_b \begin{bmatrix} k & m_a - k \\ \Omega^{-1} & \mathbf{0} \end{bmatrix} \Psi_a^H, \\ \mathbf{H}_e &= \Psi_e \Sigma_e \begin{bmatrix} k & m_a - k \\ \Omega^{-1} & \mathbf{0} \end{bmatrix} \Psi_a^H, \end{aligned} \quad (4)$$

where $\Psi_b \in \mathbb{C}^{m_b \times m_b}$, $\Psi_e \in \mathbb{C}^{m_e \times m_e}$ and $\Psi_a \in \mathbb{C}^{m_a \times m_a}$ are unitary matrices consisting of singular vectors, and Ω is a nonsingular matrix. Σ_b and Σ_e are diagonal matrices of the singular values expressed as

$$\Sigma_b = \begin{bmatrix} k-r-s & s & r \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{matrix} m_b - r - s \\ s \\ r \end{matrix} \quad (5)$$

$$\Sigma_e = \begin{bmatrix} k-r-s & s & r \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_e & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{matrix} k-r-s \\ s \\ m_e - k + r \end{matrix} \quad (6)$$

Here \mathbf{D}_b and \mathbf{D}_e are $s \times s$ diagonal matrices with elements arranged in ascending and descending orders, respectively. k , r , and s denote the total number of degrees of freedom, the number of streams serving Bob while nulling Eve, and the number of layers serving both receivers, respectively.

In [6], the precoding matrix is selected as $\mathbf{W} = \Psi_a \mathbf{B} \mathbf{P}^{1/2}$, where \mathbf{P} is a diagonal power allocation matrix, and \mathbf{B} is given by

$$\mathbf{B} = \begin{bmatrix} k & m_a - k \\ \Omega & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{matrix} k \\ m_a - k \end{matrix} \quad (7)$$

Through the use of this precoding structure, the channel is decomposed into independent layers, and power is allocated into the streams in a way to maximize the secrecy rate. The mutual information of each stream is calculated as

$$\begin{aligned} I(x_i, y_i) &= \log_2(M) \\ &- \frac{1}{M} \sum_{u=1}^M \mathbb{E} \left\{ \log_2 \left(\sum_{v=1}^M \exp \left(\frac{-f_{u,v} + \|n_i\|^2}{\sigma^2} \right) \right) \right\}, \end{aligned} \quad (8)$$

where $f_{u,v} = \|d_i \sqrt{p_i/w_i} (s_u - s_v) + n_i\|^2$, M is the cardinality of the input alphabet, \mathbf{w} is a vector obtained from diagonal elements of $\Omega^H \Omega$, and σ^2 denotes the noise variance.¹ Then, the overall secrecy rate is calculated as the sum of the individual secrecy rates of the independent streams.

In [6], determination of the power allocated to each independent stream is formulated as subproblems to maximize the individual secrecy rates. A master problem which connects the subproblems by a latent variable is used to satisfy the total power constraint, and an iterative approach is adopted to solve for the power allocations while minimizing the hidden variable. This approach is effective, however, it has a high computational complexity.

¹Eq. 8 is given in a general form, thus d_i and n_i are different for different receivers. d_i and p_i denote i -th diagonal elements of Σ and \mathbf{P} matrices, and w_i , and n_i are the i -th elements of \mathbf{w} and \mathbf{n} , respectively.

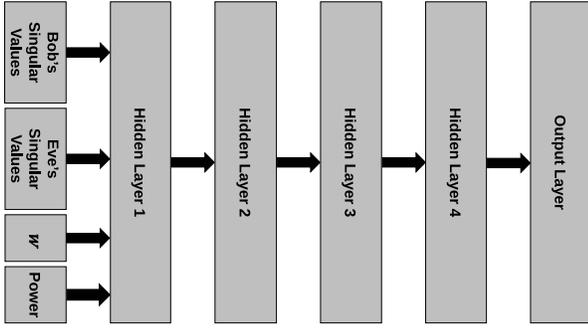


Fig. 1. DNN design for power allocation to maximize R_{sec} .

IV. POWER ALLOCATION THROUGH DNNs

As power allocation algorithm of [6] requires a large amount of computations, it may not be applicable for real-time implementation. To address this limitation, we propose a deep-learning based power allocation technique with a significantly reduced complexity.

A. Construction and Training of the DNN

The proposed DNN structure is shown in Fig. 1. It takes real-valued Σ_b , Σ_e , w and P_T as input, and tries to estimate the power allocation at the output. The number of inputs is equal to $3m_a + 1$ and the output layer has $m_a + 1$ nodes. The output dimension is taken as one more than the number of antennas because the optimal solution may require the use of less than the total available power due to the finite cardinality inputs. There are four fully-connected hidden layers in the DNN to learn the complex relationships among the singular values, w , and P_T and the power allocation vector.

Σ_b and Σ_e have their elements in the interval $[0, 1]$, however w values and the available power can be incomparably larger. In order to have a balance among different features, we normalize the power term and w by dividing them with the corresponding maximum value in the training set. The rectified linear unit (ReLU) activation function is used in the hidden layers, and the output activation is chosen as softmax. Through the use of softmax, the output values are restricted to the interval $[0, 1]$, and they sum up to 1, which is consistent with the true power allocation vector normalized with P_T .

The DNN is trained in a supervised manner through a labeled data generated with random channel realizations and the use of the iterative power allocation algorithm of [6]. The sum of the squared errors is used as the loss function to train the DNN since the power allocation is a regression problem. In order to have a generalized solution for different P_T values, the training data capture samples from a wide range of power levels of interest.

B. An Alternative Method of Training the DNN

The use of the softmax activation at the output of the DNN is a successful approach for training the neural network, however, we also consider a second alternative. In this training method, we first use ReLU activation at the output and add a custom activation function on top of it, which ensures that the

sum of the output nodes equals to P_T as expressed below

$$o_i = \frac{z_i}{\sum_{j=1}^{m_a+1} z_j} P_T \quad (9)$$

where z_i and o_i denote the i -th output node value after ReLU activation and custom activation, respectively. Through the application of the power normalization function, the output values are allowed to be in $[0, P_T]$ instead of being squeezed in $[0, 1]$. Therefore, the effects of errors for high transmit power cases are increased while keeping the labels as is (i.e., there is no normalization with P_T).

C. Complexity Analysis

In order to assess the computational complexity of the proposed solution, the testing stage is considered as the training is performed off-line and only once before the utilization of the DNN as part of a communication system. The operation in each layer of the DNN is given by $f(\mathbf{Ax} + \mathbf{b})$ where f , \mathbf{A} , \mathbf{x} , and \mathbf{b} are the nonlinear activation function, the weight matrix, the input vector, and the bias vector of the layer, respectively. That is, the computational complexity of a DNN is directly related to the number of nodes in each layer and the number of layers. In this regard, the motivation in our proposed solution is to use a DNN with a small number of nodes and layers, in an effort to reduce the computational complexity compared to the conventional iterative algorithms. The number of nodes in the input and the output layers are directly related to m_a , and the number of nodes in the hidden layers are kept in the order of the input dimensions to learn the general nonlinear pattern without increasing the complexity unnecessarily. Therefore, the size of the vectors and weight matrices may slightly change for different antenna configurations, however, they are guaranteed to be small as the number of antennas is limited.

In the conventional iterative method for determining the optimal powers, for each iteration and for each independent stream, a function $mmse(p)$ provided in [17] needs to be calculated, which requires a numerical integration and summation over the constellation points. This operation can be simplified by using a table look-up with the power levels and the corresponding MMSE values but the search for the minimum value is still time-consuming since the density of the elements in the table directly affects the convergence to the true value (for instance, approximately 5000 samples are needed for $P_T = 10$ units). Besides, a large number of iterations is needed as the step size used for updating the latent variable should be kept small for better convergence [16].

As a concrete example, consider a system with $(m_a, m_b, m_e) = (5, 5, 5)$, 16QAM symbol library, and $P_T = 10$ units. The conventional technique based on use of a table look-up requires approximately 70×10^6 comparisons, 900 additions, and 21500 multiplications to calculate the power allocation for each channel realization. On the other hand, there are four hidden layers and one output layer in the DNN as depicted in Table I, hence the proposed power allocation method requires 5 matrix multiplications, 5 vector additions, and 5 activation operations. That is, the proposed DNN requires less than 5000 multiplications,

TABLE I
DNN STRUCTURE TO ESTIMATE POWER ALLOCATION VECTOR

Layer	Number of Nodes
Input Layer	16
Hidden Layer 1	40
Hidden Layer 2	60
Hidden Layer 3	20
Hidden Layer 4	20
Output Layer	6

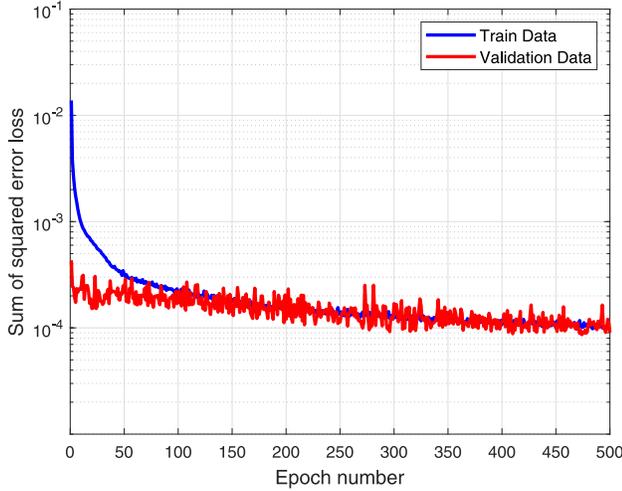


Fig. 2. Sum of the squared errors versus epochs for training and validation data (QPSK signaling, the low SNR region).

5000 additions, and 150 activation operations. Furthermore, the computational complexity of the conventional technique increases significantly when the number of antennas and available power increase, however, the complexity changes only slightly or stays the same with the proposed DNN-based solution. Note also that, the DNN can be pruned to further reduce the complexity without a loss in performance.

V. NUMERICAL RESULTS

We now present several simulation results to assess the performance of the proposed algorithm. The DNN structure in Table I is used to solve for the power allocations for different scenarios in Figs. 3 and 4. The implementation of the DNN is performed on Python by KERAS, and Adam optimizer with a learning rate of 0.005 is used [18], [19]. In the alternative DNN approach, the learning rate is chosen as 0.001 as the output values are allowed to be in $[0, P_T]$. In order to obtain a better convergence, the SNR range is split into two parts as low and high SNR regions ($[-9\text{dB}, 15\text{dB}]$ and $[12\text{dB}, 19.5\text{dB}]$), i.e., two different DNNs are employed (one for each SNR range). Training is performed for 500 epochs over the data set including 5000 samples from each SNR level chosen with 1.5dB intervals between the minimum and the maximum SNR levels. 10% of the data is utilized for the validation of the training process. It should be noted that for changing input library (e.g., with QPSK or 16QAM signaling) the DNN requires a new dataset and optimization of network variables.

Firstly, a MIMOME wiretap channel with $(m_a, m_b, m_e) = (5, 5, 5)$ is considered. As an example of the training process,

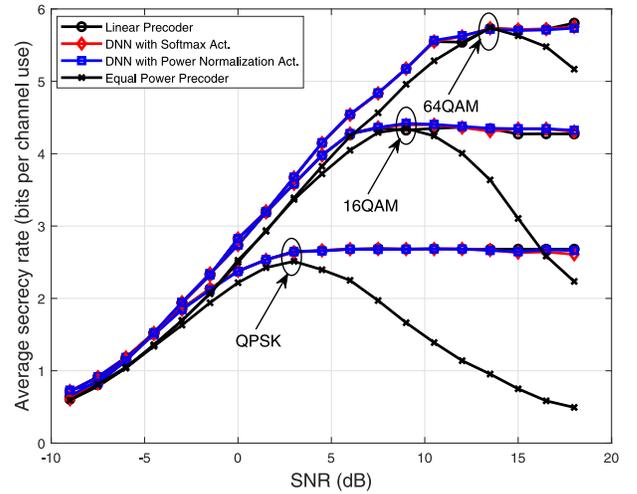


Fig. 3. Average secrecy rate versus SNR for $(m_a, m_b, m_e) = (5, 5, 5)$ configuration.

errors over the validation data and training data are given in Fig. 2. Training has not proceeded any further as the losses are small, i.e., on the order of $10^{-4} - 10^{-3}$. In Fig. 3, the trained network is used for power allocation for an independent test data including 2000 samples for each power level. The performance of the GSVD-based linear precoder with optimal power allocation, the GSVD-based equal power precoder, and the DNN-based solution with the alternative activation function are also depicted in the same figure. In the GSVD-based precoder with equal power allocation, the entire transmission power is used for transmitting streams that are more favorable to Bob (i.e., those corresponding to the diagonal elements of Σ_b that are larger than the corresponding element of Σ_e) since using a precoder favoring Eve compared to Bob is not an opportunistic strategy to improve security [6]. It is observed that the performance of the equal power precoder deteriorates rapidly in the high SNR regime since there are no streams directed only to Bob (i.e., Eve's detection capability improves through the use of excess power while Bob's performance changes only slightly as it has already reached its saturation level). It is also observed that both of the designed DNNs are capable of solving for the optimal power allocation with a small complexity.

In Fig. 4, a system with $(m_a, m_b, m_e) = (5, 5, 3)$ is considered. As the number of Eve's antennas is smaller than that of Bob, the subspace only directed to Bob is expected to be nonempty, which changes the optimal power allocation strategy, especially in the high SNR regime. That is, using all the remaining power on the streams directed only to Bob can be employed instead of using only a fraction of the total power. That is why, the performance of the equal power precoder does not decay rapidly as opposed to the previous case in which there are no streams directed only to Bob [6]. However, we still keep the dimension of the output layer the same, and allow the DNN to learn the optimal solution by the given training data.

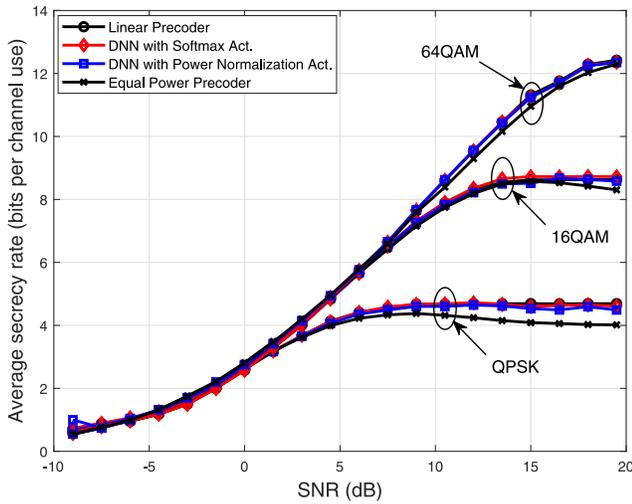


Fig. 4. Average secrecy rate versus SNR for $(m_a, m_b, m_e) = (5, 5, 3)$ configuration.

As expected, the proposed DNN manages to learn the changing behavior for the entire SNR range, and the optimal power allocation is achieved even with independent test samples.

Through the given examples, we observe that a DNN with a small number of nodes and layers can learn a general solution applicable to a wide SNR range for a highly nonlinear power allocation scheme. In the given examples, two different solutions: 1) requiring the use of only a portion of the available power, and 2) consuming the entire remaining power on streams directed only to Bob, are learned in the high SNR regime while maintaining similar solutions in the low SNR regime. Note also that we have not performed any hyperparameter optimization for the number of nodes or layers. Therefore, the number of network variables may be further reduced without any degradation in the system performance.

VI. CONCLUSION

In this letter, a power allocation strategy is considered to design a precoder to maximize the secrecy rate of a MIMOME wiretap channel with finite alphabet inputs. GSVD vectors are utilized as precoders and the required power allocation is determined by a newly developed deep neural network with a small number of nodes, instead of the computationally costly iterative algorithm. The proposed DNN is trained off-line and only once in a supervised manner by a labeled dataset containing samples from a large SNR range of interest, and it is shown that a single DNN is effective for a wide range of

SNRs. Numerical experiments show that the function approximation abilities of DNNs offer promising alternative to the iterative techniques for power allocation in precoder designs.

REFERENCES

- [1] A. D. Wyner, "The wire-tap channel," *Bell Syst. Techn. J.*, vol. 54, no. 8, pp. 1355–1387, Oct. 1975.
- [2] I. Csiszar and J. Korner, "Broadcast channels with confidential messages," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 339–348, May 1978.
- [3] A. Khisti and G. W. Wornell, "Secure transmission with multiple antennas—Part II: The MIMOME wiretap channel," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5515–5532, Nov. 2010.
- [4] S. Goel and R. Negi, "Guaranteeing secrecy using artificial noise," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2180–2189, Jun. 2008.
- [5] S. R. Aghdam and T. M. Duman, "Joint precoder and artificial noise design for MIMO wiretap channels with finite-alphabet inputs based on the cut-off rate," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3913–3923, Jun. 2017.
- [6] S. Bashar, Z. Ding, and C. Xiao, "On secrecy rate analysis of MIMO wiretap channels driven by finite-alphabet input," *IEEE Trans. Commun.*, vol. 60, no. 12, pp. 3816–3825, Dec. 2012.
- [7] Y. Wu, C. Xiao, Z. Ding, X. Gao, and S. Jin, "Linear precoding for finite-alphabet signaling over MIMOME wiretap channels," *IEEE Trans. Veh. Technol.*, vol. 61, no. 6, pp. 2599–2612, Jul. 2012.
- [8] X. Zhu, X. Zhang, W. Zeng, and J. Xie, "Deep learning-based precoder design in MIMO systems with finite-alphabet inputs," *IEEE Commun. Lett.*, vol. 24, no. 11, pp. 2518–2521, Nov. 2020.
- [9] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
- [10] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *Proc. 54th Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, Monticello, IL, USA, 2016, pp. 341–346.
- [11] S. Dörner, S. Cammerer, J. Hoydis, and S. T. Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.
- [12] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning for the Gaussian wiretap channel," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, 2019, pp. 1–6.
- [13] K. Besser, P.-H. Lin, C. R. Janda, and E. A. Jorswieck, "Wiretap code design by neural network autoencoders," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3374–3386, 2020.
- [14] L. Xiao, G. Sheng, S. Liu, H. Dai, M. Peng, and J. Song, "Deep reinforcement learning-enabled secure visible light communication against eavesdropping," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 6994–7005, Oct. 2019.
- [15] S. R. Aghdam, A. Nooraiepour, and T. M. Duman, "An overview of physical layer security with finite-alphabet signaling," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1829–1850, 2nd Quart., 2019.
- [16] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.
- [17] A. Lozano, A. M. Tulino, and S. Verdú, "Optimum power allocation for parallel Gaussian channels with arbitrary input distributions," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 3033–3051, Jul. 2006.
- [18] F. Chollet et al. (2015). *Keras*. [Online]. Available: <https://github.com/keras-team/keras>
- [19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: arXiv:1412.6980.