Deep Learning Aided Physical-Layer Security: The Security versus Reliability Trade-off

Tiep M. Hoang, Dong Liu, Thien Van Luong, Jiankang Zhang, and Lajos Hanzo

***Abstract*—This paper considers a communication system whose source can learn from channel-related data, thereby making a suitable choice of system parameters for security improvement. The security of the communication system is optimized using deep neural networks (DNNs). More explicitly, the associated security vs reliability trade-off problem is characterized in terms of the symbol error probabilities and the discrete-input continuous-output memoryless channel (DCMC) capacities. A pair of loss functions were defined by relying on the Lagrangian and on the monotonic-function based techniques. These were then used for managing the learning/training process of the DNNs for finding near-optimal solutions to the associated non-convex problem. The Lagrangian technique was shown to approach the performance of the exhaustive search. We concluded by characterizing the security vs reliability trade-off in terms of the intercept probability vs the outage probability.**

***Index terms*—**Physical layer security, reliability, deep learning, neural network, Lagrange.

I. Introduction

Securing wireless communication systems is of salient importance in the face of the ever-increasing information security threats [1]–[3]. In parallel to new network protocols, architectures and technologies, there is a vital need to conceive powerful physical-layer security (PLS) and physical-layer authentication (PLA) schemes relying on machine learning techniques [4]–[6].[1] It is believed that the introduction of machine learning to communication systems will enable them to adapt to the surrounding environment, perform cognitive functions like human intelligence, and make wise decisions to improve the quality of service [7]. One of the most powerful machine learning techniques, namely deep learning (DL), which relies on the architecture of a deep neural network (DNN), has drawn much attention from the research community [8]. However, there is a paucity of literature on applying DL to PLS solutions, which motivates this contribution.

In parallel, power-efficient system design is also an important area of research in communications [2]. For judiciously allocating and controling the transmit power of a wireless communication system, sophisticated optimization tools may be invoked for finding the optimal solutions. At the time of writing, DNNs are popular for solving stochastic optimization problems in wireless systems without deriving complex mathematical expressions [9], [10]. Moreover, DNNs have been shown to be capable of learning from the data that may have unknown time-variant distribution. A promising technique is to employ the classic Lagrangian dual method for managing the process of updating the gradients for DNNs [9], [10]. The Lagrangian dual method requires updating the *intermediate* multipliers that are not part of the DNN architecture [9], while there is another beneficial technique of solving constrained optimization problems without relying on intermediate variables [11].

A. Related Works

In the context of DL-aided PLS, the authors of [12]–[16] have used DNNs for optimizing the security. In general, the DNN-aided techniques applied for enhancing the PLS in [12]–[16] tend to aim for finding numerical solutions rather than closed-form mathematical solutions and approximate solutions (as seen in [17]). For example, the authors of [12] and [16] focus their attention on designing wiretap codes for enhancing the security level, while the works in [13]–[15] focus on power control, beamforming and artificial noise aspects, respectively. However, apart from [16], the security optimization problems of [12]–[15] only consider the average power constraint and ignore other salient constraints. Explicitly, the solution of constrained optimization problems by DNNs in the face of uncertainty has remained an open issue. More specifically, neither the Lagrangian dual method nor the so-called monotonic-function-based (MFB) method have been considered in [12]–[16].

Furthermore, when it comes to performance metrics, the secrecy rate is one of the most widely used ones. In general, the secrecy rate of a system, namely C_s, is based on the difference of the legitimate channel capacity (denoted by C_{B}) and the illegitimate channel capacity (denoted by C_{E}). In this context, most of the contributions theoretically derive C_{B} and C_{E} from the Shannon capacities by assuming that the input signal obeys a *continuous* Gaussian distribution (see [18]–[21] and references therein). However, in practice, the input signal

L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/P034284/1 and EP/P003990/1 (COALESCE) as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028).

T. M. Hoang and L. Hanzo are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mails: {tiep.hoang, lh}@soton.ac.uk).

D. Liu is with the School of Cyber Science and Technology, Beihang University, Beijing 100191, China (e-mail: dliu@buaa.edu.cn).

T. V. Luong is with the Faculty of Computer Science, Phenikaa University, Hanoi 12116, Vietnam, and also with the Phenikaa Research and Technology Institute (PRATI), A&A Green Phoenix Group JSC, Hanoi 11313, Vietnam, (e-mail: thien.luongvan@phenikaa-uni.edu.vn).

J. Zhang is with the Department of Computing and Informatics, Bournemouth University, Bournemouth BH12 5BB, U.K. (e-mail: jzhang3@bournemouth.ac.uk).

[1] The PLA focuses on identity authentication for determining users in the uplink, while the PLS basically addresses security designs, e.g. secure beamforming, in the downlink. The PLA is out of the context of this paper.

is *discrete*. Furthermore, the Shannon capacities represent the theoretical upper bounds, but not the actual capacities of channels. From a practical perspective, the discrete-input continuous-output memoryless channel (DCMC) capacities are much closer to the realistically attainable actual capacities [22]–[24], but there is only limited PLS-literature relying on the DCMC capacity. Another problem is that the symbol/bit error rates are the most popular practical performance metrics, but they are typically neglected by most of the existing PLS-related contributions. Hence, our goal is to build a bridge between the theoretical analysis and the reality by considering practical metrics. In Table I, we boldly and explicitly contrast our novel contributions to the relevant literature, which are detailed further below.

B. Contributions

Our contributions can be summarized as follows:

- We formulate a *stochastic* constrained optimization problem for striking a security vs reliability trade-off quantified in terms of the symbol error probability (SEP) vs DCMC secrecy rate. Explicitly, we find the optimal power sharing between the information-bearing and the artificial noise part.
- We then solve the problem formulated, which is stochastic non-convex optimization problem, by using deep learning. In particular, we employ the Lagrange and MFB techniques for guiding the *learning-for-optimizing* process of DNNs to find the optimal parameters. In contrast to [12]–[16], we consider not only the average power constraint, but also the security constraint. To the best of our knowledge, this is the first contribution solving this challenging constrained security optimization problem by harnessing the Lagrange as well as the MFB techniques intrinsically amalgamated with DNNs.
- Our numerical results show that the proposed deep learning-aided technique approaches the security performance that is attained by the exhaustive search, especially for the Lagrangian approach. Furthermore, we quantify the security vs reliability trade-off in terms of the intercept vs outage probabilities [25].

The remainder of the paper is organized as follows. Section II presents the system model, while Section III formulates the security-reliability trade-off in the context of constrained security optimization. In Section IV, a pair of DNN-aided techniques are conceived for solving the optimization problem considered. Our numerical results and conclusions are presented in Sections V and VI, respectively.

Notations: $\mathbb{C}^{m\times n}$ denotes the complex field that includes all complex-valued matrices of size $m\times n$; $\mathbf{I}_n$ denotes the identity matrix of size $n\times n$; The scripts $(\cdot)^\top$, $(\cdot)^*$, and $(\cdot)^\dagger$ denote the transpose, conjugate, and Hermitian operators, respectively; $\mathbf{z}\sim\mathcal{CN}(\mathbf{m},\mathbf{\Sigma})$ represents a complex Gaussian random vector with mean $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$.

II. System Model

We consider a wireless system, which consists of a source ($\mathtt{A}$), a legitimate user ($\mathtt{B}$), and a malicious eavesdropper ($\mathtt{E}$). In this system, $\mathtt{A}$ is assumed to be an intelligent entity that is aware of the environment and can learn from channel-related data to make decisions. Let us denote the instantaneous complex-valued channel between $\mathtt{A}$ and $\mathtt{B}$ at time t by $\mathbf{h}_\mathrm{B}\in\mathbb{C}^{N_\mathrm{A}\times 1}$. Furthermore, denote the instantaneous complex-valued channel between $\mathtt{A}$ and $\mathtt{E}$ at time t by $\mathbf{h}_\mathrm{E}\in\mathbb{C}^{N_\mathrm{A}\times 1}$. We also assume that the channel estimation at $\mathtt{A}$ is imperfect and only the estimate of the legitimate channel is available at $\mathtt{A}$. What $\mathtt{A}$ knows about the legitimate channel at a certain time t will be the estimated quantity $\mathbf{h}_{\mathrm{B},est}\in\mathbb{C}^{N_\mathrm{A}\times 1}$.

Then, $\mathtt{A}$ is assumed to rely on $\mathbf{h}_{\mathrm{B},est}$ for designing the beamforming vector $\mathbf{b}=\mathbf{h}^*_{\mathrm{B},est}/\|\mathbf{h}_{\mathrm{B},est}\|$. At the same time, $\mathtt{A}$ also generates the artificial noise matrix $\mathbf{A}\in\mathbb{C}^{N_\mathrm{A}\times(N_\mathrm{A}-1)}$ for drowning out the potential illegitimate users, while ensuring that $[\mathbf{b}\ \mathbf{A}]$ forms an orthonormal basis of $\mathbb{C}^{N_\mathrm{A}\times N_\mathrm{A}}$ to avoid contaminating the reception of $\mathtt{A}$. If the estimated channel $\mathbf{h}_{\mathrm{B},est}$ is exactly the same as the actual channel $\mathbf{h}_\mathrm{B}$, then the interference will become zero at $\mathtt{B}$. However, in reality, we have $\mathbf{h}_{\mathrm{B},est}\neq\mathbf{h}_\mathrm{B}$ and the relationship between $\mathbf{h}_{\mathrm{B},est}$ and $\mathbf{h}_\mathrm{B}$ can be modelled by [26], [27]

$$\mathbf{h}_\mathrm{B}=\sqrt{1-\zeta^2}\ \mathbf{h}_{\mathrm{B},est}+\zeta\ \mathbf{e}_\mathrm{B} \tag{1}$$

where $\mathbf{e}_\mathrm{B}\sim\mathcal{CN}(0,1)$ is the channel estimation error that is independent of $\mathbf{h}_{\mathrm{B},est}$, and $\zeta\in(0,1)$ denotes the estimation error coefficient. The smaller ζ, the more accurate the channel estimation. Similarly, the eavesdropping channel can be modelled as

$$\mathbf{h}_\mathrm{E}=\sqrt{1-\zeta^2}\ \mathbf{h}_{\mathrm{E},est}+\zeta\ \mathbf{e}_\mathrm{E}, \tag{2}$$

where $\mathbf{h}_{\mathrm{E},est}\sim\mathcal{CN}(0,1)$ is the estimated channel, which is known to $\mathtt{A}$, and $\mathbf{e}_\mathrm{E}\sim\mathcal{CN}(0,1)$ is the estimation error.

Combining the beamforming vector and the artificial noise matrix, $\mathtt{A}$ aims for designing a signal vector $\mathbf{s}_\mathrm{A}\in\mathbb{C}^{N_\mathrm{A}\times 1}$ as follows [28]:

$$\mathbf{s}_\mathrm{A}=\sqrt{\rho_\text{info}}\ \mathbf{b}s_\text{info}+\sqrt{\rho_\text{noise}}\ \mathbf{Az}, \tag{3}$$

where s_info is the information-bearing signal with $\mathbb{E}\left\{|s_\text{info}|^2\right\}=1$, while $\mathbf{z}\in\mathbb{C}^{(N_\mathrm{A}-1)\times 1}$ is the artificial noise with $\mathbb{E}\left\{\mathbf{zz}^\dagger\right\}=\mathbf{I}_{N_\mathrm{A}-1}$. Finally, ρ_info and ρ_noise are power-related factors. We can normalize $\mathbf{s}_\mathrm{A}$ according to $\mathbb{E}\left\{\|\mathbf{s}_\mathrm{A}\|^2\right\}=N_\mathrm{A}$, which leads to

$$\rho_\text{info}+\rho_\text{noise}(N_\mathrm{A}-1)=N_\mathrm{A}. \tag{4}$$

Due to $0\le\rho_\text{info}\le N_\mathrm{A}$, we can write $\rho_\text{info}=\varphi N_\mathrm{A}$ where $0\le\varphi\le 1$ denotes the power allocation coefficient. Accordingly, we have $\rho_\text{noise}=\frac{(1-\varphi)N_\mathrm{A}}{N_\mathrm{A}-1}$.

After $\mathtt{A}$ transmits the signal $\mathbf{s}_\mathrm{A}$, the signal received at $\mathtt{B}$ can be expressed as:

$$\begin{aligned} r_\mathrm{B}&=\sqrt{\gamma_\mathrm{A}/N_\mathrm{A}}\ \mathbf{h}_\mathrm{B}^\top\left(\sqrt{\rho_\text{info}}\mathbf{b}s_\text{info}+\sqrt{\rho_\text{noise}}\mathbf{Az}\right)+n_\mathrm{B}\\ &=\sqrt{\varphi\gamma_\mathrm{A}}\ \mathbf{h}_\mathrm{B}^\top\mathbf{b}s_\text{info}+\sqrt{\frac{(1-\varphi)\gamma_\mathrm{A}}{N_\mathrm{A}-1}}\ \mathbf{h}_\mathrm{B}^\top\mathbf{Az}+n_\mathrm{B}, \end{aligned} \tag{5}$$

where $\gamma_\mathrm{A}=\frac{P_\mathrm{A}}{\sigma^2_\mathrm{B}}$ represents the signal-to-noise ratio (SNR). Note that P_A is the average transmit power of $\mathtt{A}$, while σ^2_B is the noise variance of $\mathtt{B}$, and $n_\mathrm{B}\sim\mathcal{CN}(0,1)$ is the additive white Gaussian noise (AWGN) that is normalized.

TABLE I: Boldly contrasting our contribution to the literature

	Pre-2018 (e.g., [17])	[12] 2018	[13] 2019	[14] 2019	[16] 2020	[15] 2020	This work
Security optimization with deep learning		✓	✓	✓	✓	✓	✓
Security- or reliability-related constraints	✓				✓		✓
Deep learning with Lagrangian method							✓
Deep learning with MFB method							✓
Power control	✓		✓	✓		✓	✓
Beamforming and artificial noise	✓		✓	✓		✓	✓
Wiretap code design	✓	✓			✓		
SEP or BER	✓				✓		✓
DCMC capacity	✓						✓

Similarly, the signals received at $\mathtt{E}$ can be formulated as:

$$\begin{aligned} r_{\mathrm{E}} &= \sqrt{\gamma_{\mathrm{A}}/N_{\mathrm{A}}}\ \mathbf{h}_{\mathrm{E}}^{\top}\left(\sqrt{\rho_{\text{info}}}\mathbf{b}s_{\text{info}} + \sqrt{\rho_{\text{noise}}}\mathbf{A}\mathbf{z}\right) + n_{\mathrm{E}} \\ &= \sqrt{\varphi\gamma_{\mathrm{A}}}\ \mathbf{h}_{\mathrm{E}}^{\top}\mathbf{b}s_{\text{info}} + \sqrt{\frac{(1-\varphi)\gamma_{\mathrm{A}}}{N_{\mathrm{A}}-1}}\ \mathbf{h}_{\mathrm{E}}^{\top}\mathbf{A}\mathbf{z} + n_{\mathrm{E}}, \end{aligned} \tag{6}$$

where $n_{\mathrm{E}} \sim \mathcal{CN}\left(0, \frac{\sigma_{\mathrm{E}}^2}{\sigma_{\mathrm{B}}^2}\right)$ is the normalized AWGN, and σ_{E}^2 is the actual noise variance of $\mathtt{E}$. In general, we have $\sigma_{\mathrm{E}}^2 \neq \sigma_{\mathrm{B}}^2$. By considering the ratio $\frac{\sigma_{\mathrm{E}}^2}{\sigma_{\mathrm{B}}^2}$, we can observe that the *worst-case* scenario occurs for $\sigma_{\mathrm{E}}^2 << \sigma_{\mathrm{B}}^2$, which leads to $\frac{\sigma_{\mathrm{E}}^2}{\sigma_{\mathrm{B}}^2} \to 0$.

III. Problem Formulation

A. Symbol error probabilities

When the symbol $s_i \in \{s_1, \ldots, s_M\}$ is sent by $\mathtt{A}$ and s_j is incorrectly detected by $\mathtt{B}$, the pairwise error probability for $\mathtt{B}$, conditioned on the channel $\mathbf{h}_{\mathrm{B}}$, can be readily deduced from (5) as follows [29]:

$$\mathrm{CPEP}_{\mathrm{B}}\left(s_i \to s_j | \mathbf{h}_{\mathrm{B}}\right) = Q\left(\frac{\sqrt{\varphi\gamma_{\mathrm{A}}}|\mathbf{h}_{\mathrm{B}}^{\top}\mathbf{b}(s_i - s_j)|}{\sqrt{2\left[\frac{(1-\varphi)\gamma_{\mathrm{A}}\|\mathbf{h}_{\mathrm{B}}^{\top}\mathbf{A}\|^2}{N_{\mathrm{A}}-1} + 1\right]}}\right), \tag{7}$$

where $Q(z) = \int_z^{\infty} \frac{e^{-x^2/2}}{\sqrt{2\pi}}dx$. Similarly, the pairwise error probability for $\mathtt{E}$, conditioned on the channels $\mathbf{h}_{\mathrm{B}}$ and $\mathbf{h}_{\mathrm{E}}$, can be deduced from (6) as follows:

$$\mathrm{CPEP}_{\mathrm{E}}\left(s_i \to s_j | \mathbf{h}\right) = Q\left(\frac{\sqrt{\varphi\gamma_{\mathrm{A}}}|\mathbf{h}_{\mathrm{E}}^{\top}\mathbf{b}(s_i - s_j)|}{\sqrt{2\left[\frac{(1-\varphi)\gamma_{\mathrm{A}}\|\mathbf{h}_{\mathrm{E}}^{\top}\mathbf{A}\|^2}{N_{\mathrm{A}}-1} + \frac{\sigma_{\mathrm{E}}^2}{\sigma_{\mathrm{B}}^2}\right]}}\right), \tag{8}$$

where $\mathbf{h} = \left[\mathbf{h}_{\mathrm{B}}^{\top}, \mathbf{h}_{\mathrm{E}}^{\top}\right]^{\top}$.

The conditional symbol error probability (CSEP) for $\mathtt{B}$ and the CSEP for $\mathtt{E}$ are, respectively, given by

$$\mathrm{CSEP}_{\mathrm{B}}(\varphi|\mathbf{h}_{\mathrm{B}}) \leq \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1, j\neq i}^{M} \mathrm{CPEP}_{\mathrm{B}}\left(s_i \to s_j\right), \tag{9}$$

$$\mathrm{CSEP}_{\mathrm{E}}(\varphi|\mathbf{h}) \leq \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1, j\neq i}^{M} \mathrm{CPEP}_{\mathrm{E}}\left(s_i \to s_j\right), \tag{10}$$

where the inequality follows the rule of the union bound. Using the *nearest neighbor approximation* [29, Chapter 5], we can simplify the right hand side (RHS) of (9) by the following approximation:

$$\mathrm{CSEP}_{\mathrm{B}}(\varphi|\mathbf{h}_{\mathrm{B}}) \approx M_{d_{\min}} Q\left(\frac{\sqrt{\varphi\gamma_{\mathrm{A}}}|\mathbf{h}_{\mathrm{B}}^{\top}\mathbf{b}|d_{\min}}{\sqrt{2\left[\frac{(1-\varphi)\gamma_{\mathrm{A}}\|\mathbf{h}_{\mathrm{B}}^{\top}\mathbf{A}\|^2}{N_{\mathrm{A}}-1} + 1\right]}}\right), \tag{11}$$

where $d_{\min}$ is the minimum distance between constellation points, and $M_{d_{\min}}$ is the number of nearest neighbours at the distance $d_{\min}$. Similarly, we have

$$\mathrm{CSEP}_{\mathrm{E}}(\varphi|\mathbf{h}) \approx M_{d_{\min}} Q\left(\frac{\sqrt{\varphi\gamma_{\mathrm{A}}}|\mathbf{h}_{\mathrm{E}}^{\top}\mathbf{b}|d_{\min}}{\sqrt{2\left[\frac{(1-\varphi)\gamma_{\mathrm{A}}\|\mathbf{h}_{\mathrm{E}}^{\top}\mathbf{A}\|^2}{N_{\mathrm{A}}-1} + \frac{\sigma_{\mathrm{E}}^2}{\sigma_{\mathrm{B}}^2}\right]}}\right). \tag{12}$$

As such, the average SEP for $\mathtt{B}$ and the average SEP for $\mathtt{E}$ are given by

$$P_{\mathrm{B}} = \mathbb{E}_{\mathbf{h}_{\mathrm{B}}}\left\{\mathrm{CSEP}_{\mathrm{B}}(\varphi|\mathbf{h}_{\mathrm{B}})\right\}, \tag{13}$$

$$P_{\mathrm{E}} = \mathbb{E}_{\mathbf{h}}\left\{\mathrm{CSEP}_{\mathrm{E}}(\varphi|\mathbf{h})\right\}, \tag{14}$$

where $\mathbb{E}_{\mathbf{h}_{\mathrm{B}}}\{\cdot\}$ is the expectation over $\mathbf{h}_{\mathrm{B}}$, and $\mathbb{E}_{\mathbf{h}}\{\cdot\}$ is the expectation over $\mathbf{h} = \left[\mathbf{h}_{\mathrm{B}}^{\top}, \mathbf{h}_{\mathrm{E}}^{\top}\right]^{\top}$.

B. DCMC Secrecy Rate

Let C_{B} and C_{E} be the DCMC capacity of the $\mathtt{A}$-$\mathtt{B}$ link and that of the $\mathtt{A}$-$\mathtt{E}$ link, respectively. The DCMC secrecy rate can be defined as $C_s = [C_{\mathrm{B}} - C_{\mathrm{E}}]^+$ where $[x]^+ = x$ for $x \geq 0$ and $[x]^+ = 0$ for $x \leq 0$.

From a PLS perspective, it is worth considering the worst-case scenario, which corresponds to the lower bound of C_s. Thus, we will consider the following quantity:

$$C_s^{\text{lower}} = [C_{\mathrm{B}}^{\text{lower}} - C_{\mathrm{E}}^{\text{upper}}]^+, \tag{15}$$

where $C_{\mathrm{B}}^{\text{lower}}$ is the lower bound of C_{B}, and $C_{\mathrm{E}}^{\text{upper}}$ is the upper bound of C_{E}. Recall that the traditional PLS solutions rely on the unrealistic assumption that the input signals are continuous, e.g. Gaussian distribution [22]–[24]. By contrast, we consider a more realistic scenario where the input signals are discrete, thus formulating the secrecy rate based on the DCMC capacities but not Shannon's upper limit of channel capacity.[2]

[2] The Shannon capacity is also referred to as the continuous-input continuous-output memoryless channels (CCMC) capacity.

Proposition 1. *The expression for C_{B}^{lower} can be formulated as:*

$$C_{\mathrm{B}}^{lower} = \mathbb{E}_{\mathbf{h},\mathbf{z},n_B}\left\{g_B(\varphi|\mathbf{h},\mathbf{z},n_B)\right\}, \tag{16}$$

where

$$g_B(\varphi|\mathbf{h},\mathbf{z},n_B) = \log_2 M - \frac{1}{M}\sum_{i=1}^{M}\log_2\left[\sum_{j=1}^{M} e^{\Psi_B(i,j)}\right], \tag{17}$$

$\Psi_B(i,j) = \left(-\left|\sqrt{\varphi\gamma_A}\mathbf{h}_B^\top\mathbf{b}(s_i - s_j) + \widetilde{n_B}\right|^2 + \left|\widetilde{n_B}\right|^2\right)/\widetilde{\sigma_B}^2$, $\widetilde{n_B} = \sqrt{\frac{\gamma_A}{N_A-1}}\ \mathbf{h}_B^\top\mathbf{A}\mathbf{z} + n_B \sim \mathcal{CN}\left(0,\widetilde{\sigma_B}^2\right)$, *and* $\widetilde{\sigma_B}^2 = \frac{\gamma_A\|\mathbf{h}_B^\top\mathbf{A}\|^2}{N_A-1} + 1$. *The expectation* $\mathbb{E}_{\mathbf{h},\mathbf{z},n_B}\{\cdot\}$ *is taken over* $\mathbf{h}$, $\mathbf{z}$ *and* n_B.

Proof. See Appendix A. □

Proposition 2. *The expression for C_E^{upper} is given by*

$$C_E^{upper} = \mathbb{E}_{\mathbf{h},\mathbf{z}}\left\{g_E(\varphi|\mathbf{h},\mathbf{z})\right\}, \tag{18}$$

where

$$g_E(\varphi|\mathbf{h},\mathbf{z}) = \log_2 M - \frac{1}{M}\sum_{i=1}^{M}\log_2\left[\sum_{j=1}^{M} e^{\widetilde{\Psi_E}(i,j)}\right] \tag{19}$$

and $\widetilde{\Psi_E}(i,j) = \frac{-\left|\sqrt{\frac{\varphi(N_A-1)}{1-\varphi}}\mathbf{h}_E^\top\mathbf{b}(s_i-s_j)+\mathbf{h}_E^\top\mathbf{A}\mathbf{z}\right|^2+\left|\mathbf{h}_E^\top\mathbf{A}\mathbf{z}\right|^2}{\left\|\mathbf{h}_E^\top\mathbf{A}\right\|^2}$. *The expectation* $\mathbb{E}_{\mathbf{h},\mathbf{z}}\{\cdot\}$ *is taken over* $\mathbf{h}$ *and* $\mathbf{z}$.

Proof. See Appendix B. □

C. Stochastic Optimization Formulation

Since the estimated expressions $P_{\mathrm{B,est}}$, $P_{\mathrm{E,est}}$, $C_{\mathrm{B,est}}^{\mathrm{lower}}$ and $C_{\mathrm{E,est}}^{\mathrm{upper}}$ are available at A, they will be employed for optimizing the security performance of the proposed system. In particular, A now aims for finding φ so that

$$\begin{cases} P_{\mathrm{B}} \text{ is minimized} \\ P_{\mathrm{E}} \text{ is maximized} \end{cases} \Rightarrow \begin{cases} P_{\mathrm{B}} \text{ is minimized} \\ |P_{\mathrm{E,max}} - P_{\mathrm{E}}| \text{ is minimized} \end{cases}$$
$$\Rightarrow \begin{cases} \mathbb{E}_{\mathbf{h}}\left\{\mathrm{CSEP_B}\right\} \text{ is minimized} \\ \mathbb{E}_{\mathbf{h}}\left\{|P_{\mathrm{E,max}} - \mathrm{CSEP_E}|\right\} \text{ is minimized} \end{cases} \tag{20}$$

where $P_{\mathrm{E,max}} = \frac{M-1}{M}$ is the maximum SEP of E. In general, it is a challenging goal to minimize any multi-component objective function (OF), which will lead to a multiple-objective optimization (MOO) problem. Explicitly, our optimization objective of (20) can be formulated in the form of linear scalarization as follows:

$$\min_{\varphi} \quad \epsilon P_{\mathrm{B}} + \widehat{\epsilon}(P_{\mathrm{E,max}} - P_{\mathrm{E}}), \tag{21a}$$

where $\epsilon \in [0,1]$ is the *priority* factor, and $\widehat{\epsilon} = 1 - \epsilon$. If $\epsilon = 1$, we only minimize the average SEP of B without considering the average SEP of E. By contrast, $\epsilon = 0$ means that we only minimize the average SEP of E without considering the average SEP of B. Hence, the value of ϵ indicates our priority and helps in striking a trade-off between the above pair of conflicting objectives.

To guarantee the security level of our proposed system, solving the MOO problem (21) may not be sufficient by itself. Since C_s^{lower} is based on the difference between $C_{\mathrm{B}}^{\mathrm{lower}}$ and $C_{\mathrm{E}}^{\mathrm{upper}}$, we have to consider the following constraint:

$$C_{\mathrm{B}}^{\mathrm{lower}} - C_{\mathrm{E}}^{\mathrm{upper}} \geq 0. \tag{22}$$

Upon combining (21) and (22) together, we arrive at the following stochastic optimization problem:

$$\min_{\varphi} \quad \epsilon\,\mathbb{E}_{\mathbf{h}}\left\{\mathrm{CSEP_B}\right\} + \widehat{\epsilon}\,\mathbb{E}_{\mathbf{h}}\left\{|P_{\mathrm{E,max}} - \mathrm{CSEP_E}|\right\}, \tag{23a}$$
$$\text{s.t.} \quad C_{\mathrm{B}}^{\mathrm{lower}} - C_{\mathrm{E}}^{\mathrm{upper}} \geq 0. \tag{23b}$$

Substituting (11), (12), (16) and (18) into (23), we can rewrite (23) as follows:

$$\min_{\varphi} \quad \mathbb{E}_{\mathbf{h}}\left\{\mathcal{F}(\varphi)\right\}, \tag{24a}$$
$$\text{s.t.} \quad \mathbb{E}_{\mathbf{h},n_{\mathrm{B}},\mathbf{z}}\left\{\mathcal{C}(\varphi)\right\} \leq 0, \tag{24b}$$

where the functions $\mathcal{F}(\varphi)$ and $\mathcal{C}(\varphi)$ are defined as

$$\mathcal{F}(\varphi) \triangleq \epsilon M_{d_{\min}} Q\left(\frac{\sqrt{\varphi\gamma_{\mathrm{A}}}\|\mathbf{h}_{\mathrm{B}}\|d_{\min}}{\sqrt{2}}\right) + \widehat{\epsilon}\left|\frac{M-1}{M} - M_{d_{\min}} Q\left(\frac{\sqrt{\varphi\gamma_{\mathrm{A}}}|\mathbf{h}_{\mathrm{E}}^\top\mathbf{b}|d_{\min}}{\sqrt{2\left[\frac{(1-\varphi)\gamma_{\mathrm{A}}\|\mathbf{h}_{\mathrm{E}}^\top\mathbf{A}\|^2}{N_{\mathrm{A}}-1} + \frac{\sigma_{\mathrm{E}}^2}{\sigma_{\mathrm{B}}^2}\right]}}\right)\right|, \tag{25}$$

and

$$\mathcal{C}(\varphi) = \sum_{i=1}^{M}\log_2\left[\frac{\sum_{j=1}^{M} e^{\Psi_{\mathrm{B}}(i,j)}}{\sum_{j=1}^{M} e^{\widetilde{\Psi_{\mathrm{E}}}(i,j)}}\right]. \tag{26}$$

To elaborate, the stochastic optimization problem (24) is non-convex, hence it is hard to solve by conventional optimization techniques. Thus, we will tackle this challenge by using powerful DNNs. In the sequel, a pair of DNN-based approaches will be proposed to handle the stochastic non-convex optimization problem (24).

IV. Deep-Learning-aided Security Designs

This section presents a pair of alternative techniques of solving (24) (or equivalently (23)). As an intelligent entity, which can learn from the statistical history of channel-related data to decide system parameters for improving security, A will employ DNNs to achieve this goal.

There are two types of DNNs to be considered in this work. As for the first type, we use the Lagrangian dual method for carrying out the gradient updates on primal and dual variables, and then we set the Lagrange function to be the loss function of a DNN. As for the second technique, we formulate a cost function based on a certain monotonic function and manage the learning process through directly computing the gradient of the cost function. In general, the main difference between these two types of DNNs lies in the formulation of the loss function. In both approaches, the gradient updates are performed with the aid of unsupervised DNNs. Note that we will use the term "Lagrange-DNNs" to refer to the DNNs used for the Lagrangian technique. Similarly, the terms "MFB-DNNs" will

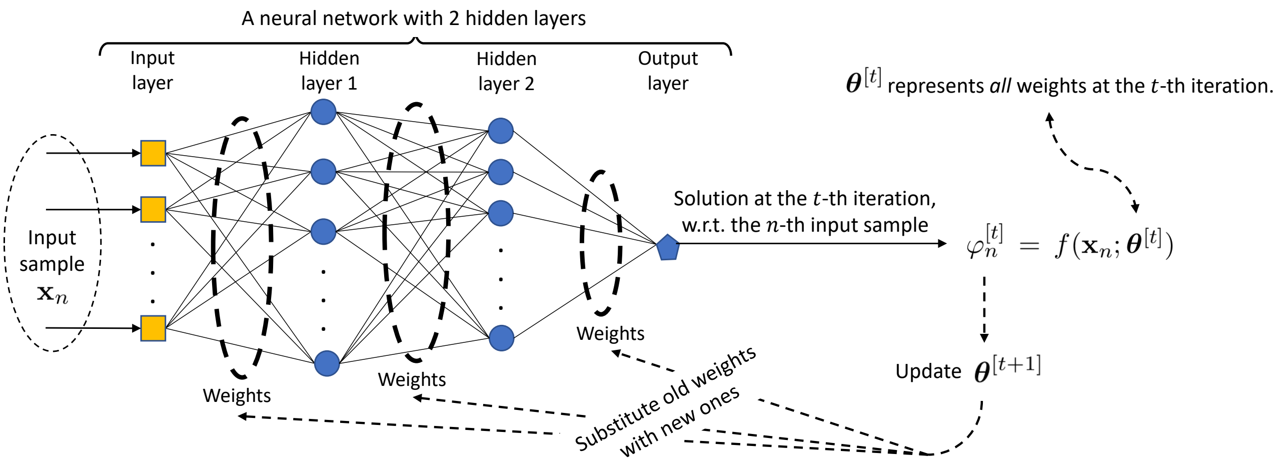

A neural network with 2 hidden layers
Input layer
Hidden layer 1
Hidden layer 2
Output layer
Input sample $\mathbf{x}_n$
Weights
Weights
Weights
Solution at the t-th iteration, w.r.t. the n-th input sample
$\varphi_n^{[t]} = f(\mathbf{x}_n; \boldsymbol{\theta}^{[t]})$
$\boldsymbol{\theta}^{[t]}$ represents *all* weights at the t-th iteration.
Update $\boldsymbol{\theta}^{[t+1]}$
Substitute old weights with new ones

Fig. 1: The architecture of a DNN and the weight updating rule are illustrated.

refer to the DNNs used for the second approach, where MFB stands for “monotonic-function-based”.

The Lagrange-DNNs will be presented in Sub-section IV-B, while the MFB-DNNs will be presented in Sub-section IV-C. Prior to delving into the Lagrange-DNNs and the MFB-DNNs, we will discuss the gradient updating mechanism of a generic DNN in Sub-section IV-A.

A. Gradient Descent and Approximation by DNNs

Let $f_{\boldsymbol{\theta}}$ represent a function parameterized by a certain vector $\boldsymbol{\theta}$. Then, the mapping $f_{\boldsymbol{\theta}} : \mathbf{x} \rightarrow \varphi$ characterizes the functional relationship between the input $\mathbf{x}$ and the output φ, formulated as $\varphi = f(\mathbf{x}; \boldsymbol{\theta})$. Furthermore, we define $\ell(\varphi)$ as a function of φ and call $\ell(\varphi)$ a loss/cost/risk function. The physical meanings of $\mathbf{x}$, φ and $\boldsymbol{\theta}$ can be interpreted as follows:

- $\mathbf{x}$ is the channel-related data and obeys some distribution. Considering the sampling over time, we have $\mathbf{x}_n$ as the realization of $\mathbf{x}$ at a certain time slot n. After $|\mathcal{B}|$ time slots, we obtain a batch of samples, denoted by $\mathcal{B} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|\mathcal{B}|}\}$. Note that $|\mathcal{B}|$ is the cardinality of $\mathcal{B}$.
- $\boldsymbol{\theta}$ is not invariant, but it can be *iteratively* updated in order to improve the relationship $f_{\boldsymbol{\theta}}$ between $\mathbf{x}$ and φ. At the t-th iteration, we have $\boldsymbol{\theta} = \boldsymbol{\theta}^{[t]}$.
- φ is the power allocation coefficient, which needs to be optimized. φ will be chosen so that the loss function is minimized. With respect to the n-th input sample $\mathbf{x}_n$, the specific value of the output φ at the t-th iteration is $\varphi_n^{[t]}$. For notational simplicity, we will denote
$$\varphi_n = f(\mathbf{x}; \boldsymbol{\theta})\big|_{\mathbf{x}=\mathbf{x}_n} = f(\mathbf{x}_n; \boldsymbol{\theta})$$
as a simplified version of the function φ, when $\mathbf{x}$ is substituted by a specific value $\mathbf{x}_n$. Also, let us denote
$$\varphi_n^{[t]} = f(\mathbf{x}; \boldsymbol{\theta})\big|_{\mathbf{x}=\mathbf{x}_n,\ \boldsymbol{\theta}=\boldsymbol{\theta}^{[t]}} = f(\mathbf{x}_n; \boldsymbol{\theta}^{[t]})$$
as a specific value of the function φ, when respectively substituting $\mathbf{x}$ and $\boldsymbol{\theta}$ by specific values $\mathbf{x}_n$ and $\boldsymbol{\theta}^{[t]}$. Obviously, we have $\varphi_n^{[t]} = \varphi_n\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{[t]}}$.

As $\mathbf{x}$ is random, it is worth considering the average quantity
$$\mathcal{L}(\varphi) = \mathbb{E}\left\{\ell\left(\varphi\right)\right\} = \mathbb{E}_{\mathbf{x}}\left\{\ell\left(f_{\boldsymbol{\theta}}\right)\right\}. \tag{27}$$
To lower the average loss $\mathcal{L}(\varphi)$, we aim to solve the following optimization problem:
$$\min_{\varphi} \mathcal{L}(\varphi) \Leftrightarrow \min_{f_{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{x}}\left\{\ell\left(f_{\boldsymbol{\theta}}\right)\right\} \Leftrightarrow \min_{f_{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{x}}\left\{\ell\big(f(\mathbf{x}; \boldsymbol{\theta})\big)\right\}. \tag{28}$$
In practice, we may not have sufficiently statistical knowledge of the truth distribution of $\mathbf{x}$. Instead, we may only have a batch $\mathcal{B}$ of data samples. Using the data batch $\mathcal{B}$, we can relax the problem (28) as follows:
$$\min_{\varphi} \mathcal{L}(\varphi) \approx \min_{f_{\boldsymbol{\theta}}} \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x}_n \in \mathcal{B}} \ell\big(f(\mathbf{x}_n; \boldsymbol{\theta})\big). \tag{29}$$
The relationship between (29) and our proposed problem (24) will be discussed in detail in the next sub-sections. For now, let us just focus on how to use DNNs for solving (29) generally.

Using the gradient descent for finding the solution to (29), we can update the parameter vector $\boldsymbol{\theta}$ according to the following rule:
$$\boldsymbol{\theta}^{[t+1]} = \boldsymbol{\theta}^{[t]} - \frac{\eta}{|\mathcal{B}|} \sum_{\mathbf{x}_n \in \mathcal{B}} \nabla_{\boldsymbol{\theta}} \ell\big(\varphi_n\big)\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{[t]}}. \tag{30}$$
In order to perform the above weight update with the help of a DNN, we will comply with the following designs:

- A DNN takes $\mathbf{x}$ as the input and outputs φ as the solution.
- The weights of a DNN are the elements in $\boldsymbol{\theta}$.

As a result, the weight update in (30) is equivalent to the process of training/updating the DNN weights. Figure 1 illustrates the architecture of a DNN consisting of two hidden layers.

In short, the DNN training process aims to update the weight vector $\boldsymbol{\theta}$ in order to improve the functional relationship $f_{\boldsymbol{\theta}}$ between the channel-related data $\mathbf{x}$ and the power allocation policy φ. Noticeably, a DNN consists of connected neurons (as illustrated in Figure 1), thus the functional relationship $f_{\boldsymbol{\theta}}$ can be further improved by combining neurons together. Although a single neuron is activated by a simple function, the connection of multiple neurons will form a more complex function, whereby enabling us to learn the mapping $f_{\boldsymbol{\theta}}$ as best as possible. According to [10], [30], [31], DNNs are capable of approximating sophisticated functions with any tolerance.

Remark 1. *The DNN architecture seen in Figure 1 depicts the relationship $f_{\boldsymbol{\theta}} : \mathbf{x} \to \varphi$ between the data $\mathbf{x}$ and the variable φ. However, the output layer of this DNN architecture can be re-designed to cope with multivariate problems. If we want to minimize a certain function of the form $\mathcal{L}(\varphi, \omega, \phi)$, then we can re-design the output layer with more than one neuron in order to find many functional relationships between the data and the variables φ, ω, and ϕ. For example, if ω is complex-valued and ϕ is real-valued, then the output layer can have 4 neurons, which correspond to the following functional relationships: $f_{\boldsymbol{\theta}} : \mathbf{x} \to \varphi$, $\check{f}_{\boldsymbol{\theta}} : \mathbf{x} \to \Re\{\omega\}$, $\hat{f}_{\boldsymbol{\theta}} : \mathbf{x} \to \Im\{\omega\}$, and $\bar{f}_{\boldsymbol{\theta}} : \mathbf{x} \to \phi$.*

In the following subsections, two different types of DNNs will be considered: i) Lagrange-DNNs and ii) MFB-DNNs. Both these two types apply the gradient descent to update their weights. In other words, the weight updating mechanism (30) can be applied to both the Lagrange-DNN and the MFB-DNN. To distinguish the notations w.r.t. Lagrange-DNNs from the notations w.r.t. MFB-DNNs, we will consider the following alternative notations:

$$\boldsymbol{\theta}, \boldsymbol{\theta}^{[t]} \xrightarrow[\quad]{\text{replaced by}} \begin{cases} \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\theta}}^{[t]}, & \text{w.r.t. Lagrange-DNNs;} \\ \widetilde{\boldsymbol{\theta}}, \widetilde{\boldsymbol{\theta}}^{[t]}, & \text{w.r.t. MFB-DNNs,} \end{cases}$$

$$\varphi, \varphi_n, \varphi_n^{[t]} \xrightarrow[\quad]{\text{replaced by}} \begin{cases} \widehat{\varphi}, \widehat{\varphi}_n, \widehat{\varphi}_n^{[t]}, & \text{w.r.t. Lagrange-DNNs;} \\ \widetilde{\varphi}, \widetilde{\varphi}_n, \widetilde{\varphi}_n^{[t]}, & \text{w.r.t. MFB-DNNs.} \end{cases}$$

B. Lagrange Approach

This sub-section presents how to employ the Lagrangian dual method for managing the gradient update of Lagrange-DNNs during the learning process. First, the average loss function $\mathcal{L}(\widehat{\varphi})$ in (27) is explicitly formulated as follows:

$$\begin{aligned} \mathcal{L}(\widehat{\varphi}) &= \mathbb{E}_{\mathbf{h}, n_{\mathrm{B}}, \mathbf{z}} \left\{ \mathcal{F}(\widehat{\varphi}) \right\} + \xi_{\text{dual}} \, \mathbb{E}_{\mathbf{h}, n_{\mathrm{B}}, \mathbf{z}} \left\{ \mathcal{C}(\widehat{\varphi}) \right\} \\ &\triangleq \mathcal{L}^{\text{Lag}}(\xi_{\text{dual}}, \widehat{\varphi}), \end{aligned} \tag{31}$$

where ξ_{dual} is the Lagrange multiplier associated with the stochastic constraint (24b). Note that $\mathcal{L}^{\text{Lag}}(\xi_{\text{dual}}, \widehat{\varphi})$ is also called the Lagrange function. Then, using the Lagrangian dual method [32], we can convert (24) into the following primal-dual problem:

$$\max_{\xi_{\text{dual}} \geq 0} \;\; \min_{\widehat{\varphi}} \;\; \mathcal{L}^{\text{Lag}}(\xi_{\text{dual}}, \widehat{\varphi}), \tag{32}$$

It was shown in [9], [10] that Lagrange-DNNs can be employed for solving the primal-dual problems. To be more specific, for the problem (32), instead of directly finding the optimal value $\widehat{\varphi}^*$ of $\widehat{\varphi}$, we can use a certain Lagrange-DNN for outputting $\widehat{\varphi}_n^{[t]} \approx \widehat{\varphi}^*$. This means that the Lagrange-DNN is used for learning the functional relationship $f_{\widehat{\boldsymbol{\theta}}} : \mathbf{x} \to \widehat{\varphi}$ so that the average loss function $\mathcal{L}^{\text{Lag}}(\xi_{\text{dual}}, f_{\widehat{\boldsymbol{\theta}}})$ is minimized. Recall that $\widehat{\boldsymbol{\theta}}$ denotes the vector capturing all the weights of the Lagrange-DNN. Note that once the Lagrange-DNN has been employed, the Lagrange dual problem (32) becomes the following approximate problem:

$$\max_{\xi_{\text{dual}} \geq 0} \; \min_{\widehat{\boldsymbol{\theta}}} \;\; \mathcal{L}^{\text{Lag}} \left(\xi_{\text{dual}}, f_{\widehat{\boldsymbol{\theta}}} \right). \tag{33}$$

Algorithm 1: Using DNNs for solving the problem (33)

```
1: Initialize $\xi_{\text{dual}}^{[0]} \geq 0$ for $\xi_{\text{dual}}$.
2: Train a DNN to minimize $\mathcal{L}^{\text{Lag}}\left(\xi_{\text{dual}}, f_{\widehat{\boldsymbol{\theta}}}\right)$.
3: Upon convergence at the $(t+1)$-th epoch, get a critical
   point $\varphi^*_{(\xi_{\text{dual}}>0)} \triangleq f(\widehat{\boldsymbol{\theta}}^{[t+1]})$.
4: **if** $\xi_{\text{dual}}$ is less than a small tolerance, e.g., $\xi_{\text{dual}} \leq 10^{-6}$,
   **then**
6:     $\xi_{\text{dual}}$ is considered as zero. Assign the desired solution
       $\varphi^* \longleftarrow \varphi^*_{(\xi_{\text{dual}}>0)}$.
7:     Go to the last step in line 15.
8: **else**
10:    Train another DNN to minimize $\mathcal{L}^{\text{Lag}}(\xi_{\text{dual}} = 0, f_{\widehat{\boldsymbol{\theta}}})$.
11:    Upon convergence at the $(t+1)$-th epoch, get another
       critical point $\varphi^*_{(\xi_{\text{dual}}=0)} \triangleq f(\widehat{\boldsymbol{\theta}}^{[t+1]})$.
12:    Compare the critical points $\varphi^*_{(\xi_{\text{dual}}>0)}$ and $\varphi^*_{(\xi_{\text{dual}}=0)}$ to
       select the optimal solution $\varphi^*$ to the problem (24).
       Assign the desired solution $\varphi^* \longleftarrow \varphi^*_{(\xi_{\text{dual}}>0)}$, only if
       $\varphi^*_{(\xi_{\text{dual}}=0)}$ is a better solution than $\varphi^*_{(\xi_{\text{dual}}>0)}$.
13:    Go to the last step in line 15.
14: **end if**
15: Return $\varphi^*$ as the desired solution.
```

Due to the relationship $\widehat{\varphi}_n^{[t]} \approx \widehat{\varphi}^*$, the solution of (33) constitutes an approximation of the *actual* optimal solution of (32). However, the optimality gap between these two solutions can be arbitrarily small, thanks to the universal approximation theorem of [10], [30].

On the basis of the Lagrangian dual method, the primal variable $\widehat{\boldsymbol{\theta}}$ and the dual variable ξ_{dual} can be iteratively updated. For the primal update, at the $(t+1)$-st iteration, we have

$$\begin{aligned} \widehat{\boldsymbol{\theta}}^{[t+1]} = \widehat{\boldsymbol{\theta}}^{[t]} - \eta \Bigg[& \nabla_{\widehat{\boldsymbol{\theta}}} \mathbb{E} \left\{ \mathcal{F}\left(f_{\widehat{\boldsymbol{\theta}}}\right) \right\} \Big|_{\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}^{[t]}} \\ & + \xi_{\text{dual}}^{[t]} \nabla_{\widehat{\boldsymbol{\theta}}} \mathbb{E} \left\{ \mathcal{C}\left(f_{\widehat{\boldsymbol{\theta}}}\right) \right\} \Big|_{\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}^{[t]}} \Bigg]. \end{aligned} \tag{34}$$

For the dual update, at the $(t+1)$-st iteration, we have

$$\xi_{\text{dual}}^{[t+1]} = \left[\xi_{\text{dual}}^{[t]} + \eta \, \mathbb{E} \left\{ \mathcal{C}\left(f_{\widehat{\boldsymbol{\theta}}}\right) \right\} \Big|_{\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}^{[t]}} \right]^+ . \tag{35}$$

In (35), the operator $[\cdot]^+$ is used for ensuring that ξ_{dual} is non-negative in any iteration. Note that $\widehat{\boldsymbol{\theta}}$ is the weight vector of the Lagrange-DNN in this Lagrangian technique, thus the primal update is performed within the Lagrange-DNN. By contrast, the dual update of ξ_{dual} is performed outside the Lagrange-DNN.

The expectation operator $\mathbb{E}_{\mathbf{h}, n_{\mathrm{B}}, \mathbf{z}}\{\cdot\}$ in (34), as well as in (35), represents averaging over the entire training dataset; however, it is computationally more attractive to perform the expectation over a smaller set of examples, namely, a mini batch $\mathcal{B}$. In doing so, it may be readily shown that the nature of DNNs may be efficiently exploited by using the stochastic gradient descent (SGD) method [33].[3] In this vein, the primal

[3] Conventional gradient-based techniques without a DNN cannot learn from the data. By contrast, the integration of DNNs will help a system to learn non-linear functions.

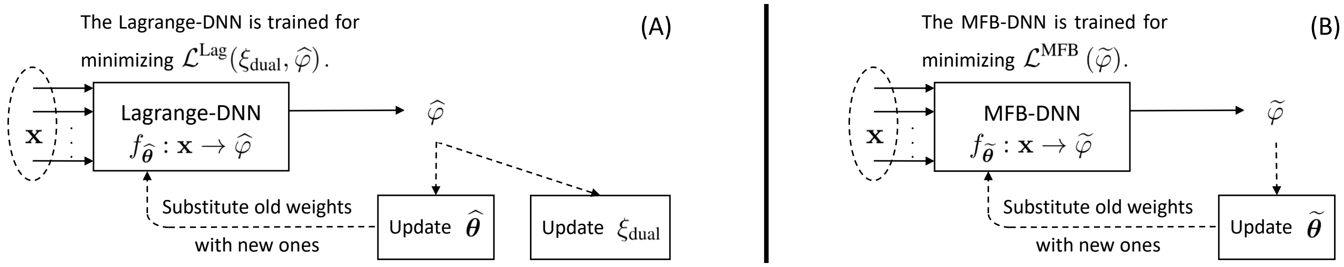

The Lagrange-DNN is trained for minimizing $\mathcal{L}^{\text{Lag}}(\xi_{\text{dual}}, \widehat{\varphi})$.
(A)
$\mathbf{x}$
Lagrange-DNN
$f_{\widehat{\boldsymbol{\theta}}} : \mathbf{x} \to \widehat{\varphi}$
$\widehat{\varphi}$
Substitute old weights with new ones
Update $\widehat{\boldsymbol{\theta}}$
Update ξ_{dual}
The MFB-DNN is trained for minimizing $\mathcal{L}^{\text{MFB}}(\widetilde{\varphi})$.
(B)
$\mathbf{x}$
MFB-DNN
$f_{\widetilde{\boldsymbol{\theta}}} : \mathbf{x} \to \widetilde{\varphi}$
$\widetilde{\varphi}$
Substitute old weights with new ones
Update $\widetilde{\boldsymbol{\theta}}$

Fig. 2: The Lagrange approach is depicted in the sub-figure (A), while the MFB approach is depicted in the sub-figure (B).

and dual updates in (34) and (35) will be, respectively, adjusted into

$$\widehat{\boldsymbol{\theta}}^{[t+1]} = \widehat{\boldsymbol{\theta}}^{[t]} - \frac{\eta}{|\mathcal{B}|}\sum_{n=1}^{|\mathcal{B}|}\Bigg\{\Bigg[\nabla_{\widehat{\varphi}_n}\mathcal{F}\left(\widehat{\varphi}_n\right)\Big|_{\widehat{\varphi}_n=\widehat{\varphi}_n^{[t]}} + \xi_{\text{dual}}^{[t]}\nabla_{\widehat{\varphi}_n}\mathcal{C}\left(\widehat{\varphi}_n\right)\Big|_{\widehat{\varphi}_n=\widehat{\varphi}_n^{[t]}}\Bigg] \times \Bigg[\nabla_{\widehat{\boldsymbol{\theta}}}\widehat{\varphi}_n\Big|_{\widehat{\boldsymbol{\theta}}=\widehat{\boldsymbol{\theta}}^{[t]}}\Bigg]\Bigg\} \quad (36)$$

and

$$\xi_{\text{dual}}^{[t+1]} = \left[\xi_{\text{dual}}^{[t]} + \frac{\eta}{|\mathcal{B}|}\sum_{n=1}^{|\mathcal{B}|}\mathcal{C}\left(\widehat{\varphi}_n^{[t]}\right)\right]^{+}. \quad (37)$$

Herein, $|\mathcal{B}|$ denotes the size of the mini batch $\mathcal{B}$, which has to be sufficiently large for guaranteeing the convergence of (36) and (37).

In short, the process of handling the Lagrangian dual problem (33) can be summarized in Algorithm 1. It should be noted that the Steps 10-13 of Algorithm 1 ensure that if a critical point of the Lagrangian function $\mathcal{L}^{\text{Lag}}\left(\xi_{\text{dual}}, f_{\widehat{\boldsymbol{\theta}}}\right)$ is associated with the multiplier $\xi_{\text{dual}} = 0$, we will not miss it. Although the dual update at the $(t+1)$-st iteration can lead to ξ_{dual} being equal to a small positive number $\xi_{\text{dual}}^{[t+1]}$, this value may still be considered as relatively large, given that the OF is constituted by the SEPs that may have very low values. Another reason is that a Lagrange-DNN may encounter the issue of having a vanishing gradient, which prevents that Lagrange-DNN from getting an update, thus yielding a $\xi_{\text{dual}}^{[t+1]}$ value far away from the actual optimal value of ξ_{dual}.

C. Monotonic-function-based (MFB) approach

In contrast to the cost function $\mathcal{L}^{\text{Lag}}(\xi_{\text{dual}}, \widehat{\varphi})$ of the Lagrangian approach, we will formulate another cost function that incorporates the stochastic constraint (24b). In this vein, we use another DNN, namely the MFB-DNN, whose weight vector and output are denoted by $\widetilde{\boldsymbol{\theta}}$ and $\widetilde{\varphi}$, respectively.

When using the MFB-DNN, we aim to find the functional relationship $f_{\widetilde{\boldsymbol{\theta}}} : \mathbf{x} \to \widetilde{\varphi}$ so that the average loss function $\mathcal{L}(\widetilde{\varphi})$ is minimized. In other words, the MFB-DNN is employed for outputting $\widetilde{\varphi}_n^{[t]} \approx \widetilde{\varphi}^{*}$. Different from the formulation of the loss function $\mathcal{L}\left(\widehat{\varphi}\right) = \mathcal{L}^{\text{Lag}}(\xi_{\text{dual}}, \widehat{\varphi})$ in the Lagrange approach, the loss function of the MFB-DNN in the MFB method will be formulated as:

$$\begin{aligned}\mathcal{L}\left(\widetilde{\varphi}\right) &= \mathcal{L}^{\text{MFB}}\left(\widetilde{\varphi}\right) \\ &\triangleq \lambda_1\,\mathbb{E}_{\mathbf{h}}\left\{\mathcal{F}\left(\widetilde{\varphi}\right)\right\} \\ &\quad + (1-\lambda_1)\mathcal{M}\left(\left[\frac{1}{\lambda_2}\,\mathbb{E}_{\mathbf{h}}\left\{\mathcal{C}\left(\widetilde{\varphi}\right)\right\}\right]^{+}\right),\end{aligned} \quad (38)$$

where $0 < \lambda_1 < 1$ is the factor balancing the OF $\mathbb{E}_{\mathbf{h}}\left\{\mathcal{F}\left(\widetilde{\varphi}\right)\right\}$ and the left hand side $\mathbb{E}_{\mathbf{h}}\left\{\mathcal{C}\left(\widetilde{\varphi}\right)\right\}$ of the constraint; $\lambda_2 > 0$ is the scaling factor; and $\mathcal{M}([z]^{+})$ can be one of the popular *monotonic* functions like the ReLU function, sigmoid function, tanh function and sofplus function.[4] Based on the fact that $\mathcal{M}[\max(0,z)]$ is a *monotonically increasing* function of z, we observe that

$$\begin{cases}\mathcal{M}(\max(0,z)) = \mathcal{M}(0) \geq 0 & \text{when } z \leq 0 \\ \mathcal{M}(\max(0,z)) = \mathcal{M}(z) > \mathcal{M}(0) & \text{when } z > 0\end{cases}. \quad (39)$$

As such, $\mathcal{M}[\max(0,z)]$ reaches the minimum value $\mathcal{M}(0)$ when $z \leq 0$. Likewise, the second term in (38) reaches the minimum value $\mathcal{M}(0)$ when $\mathbb{E}_{\mathbf{h}}\left\{\mathcal{C}(\widetilde{\varphi})\right\} \leq 0$, i.e.,

$$\begin{aligned}&\mathcal{M}\left(\frac{1}{\lambda_2}\left[\mathbb{E}_{\mathbf{h}}\left\{\mathcal{C}\left(\widetilde{\varphi}\right)\right\}\right]^{+}\right) = \mathcal{M}(0) \\ &\Longleftrightarrow \mathbb{E}_{\mathbf{h}}\left\{\mathcal{C}\left(\widetilde{\varphi}\right)\right\} \leq 0.\end{aligned} \quad (40)$$

From (38)–(39), we can deduce the following inequality:

$$\mathcal{L}^{\text{MFB}}\left(\widetilde{\varphi}\right) \geq -\lambda_1\,\mathbb{E}_{\mathbf{h}}\left\{\mathcal{F}\left(\widetilde{\varphi}\right)\right\} + (1-\lambda_1)\mathcal{M}(0). \quad (41)$$

Ideally, $\mathcal{L}^{\text{MFB}}\left(\widetilde{\varphi}\right)$ is minimized when the equality "=" occurs in (41). In this ideal case, minimizing $\mathcal{L}^{\text{MFB}}\left(\widetilde{\varphi}\right)$ corresponds to minimizing the OF $\mathbb{E}_{\mathbf{h}}\left\{\mathcal{F}\left(\widetilde{\varphi}\right)\right\}$. Moreover, the occurrence of the equality also corresponds to the occurrence of (40), i.e. the stochastic constraint of (24b) is approximately satisfied.

Although, the equality in (41) may not always be achievable, it is promising that the process of training a neural network to minimize $\mathcal{L}^{\text{MFB}}\left(\widetilde{\varphi}_n\right)$ might find the solution of the original problem (24). Replacing $\widetilde{\varphi}_n$ by $f_{\widetilde{\boldsymbol{\theta}}}$ and applying the gradient

[4] In simulation, we use $\mathcal{M}[\max(0,z)] = \log_2(1 + e^{\max(0,z)})$, which is based on the softplus function.

descent, the weight vector $\widetilde{\boldsymbol{\theta}}$ of the MFB-DNN can be iteratively updated as follows:

$$\begin{aligned}\widetilde{\boldsymbol{\theta}}^{[t+1]} &= \widetilde{\boldsymbol{\theta}}^{[t]} - \eta \nabla_{\widetilde{\boldsymbol{\theta}}} \mathcal{L}^{\text{MFB}} \left(f_{\widetilde{\boldsymbol{\theta}}}\right)\Big|_{\widetilde{\boldsymbol{\theta}}=\widetilde{\boldsymbol{\theta}}^{[t]}} \\ &= \widetilde{\boldsymbol{\theta}}^{[t]} - \eta \Bigg[\lambda_1 \nabla_{\widetilde{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{h}} \left\{ \mathcal{F}\left(f_{\widetilde{\boldsymbol{\theta}}}\right) \right\} \Big|_{\widetilde{\boldsymbol{\theta}}=\widetilde{\boldsymbol{\theta}}^{[t]}} \\ &\qquad + (1-\lambda_1) \\ &\qquad \times \nabla_{\widetilde{\boldsymbol{\theta}}} \mathcal{M} \left(\frac{\left[\mathbb{E}_{\mathbf{h}} \left\{ \mathcal{C}\left(f_{\widetilde{\boldsymbol{\theta}}}\right)\right\}\right]^{+}}{\lambda_2} \right) \Bigg|_{\widetilde{\boldsymbol{\theta}}=\widetilde{\boldsymbol{\theta}}^{[t]}} \Bigg]. \end{aligned} \tag{42}$$

It is plausible for the MFB-DNN that it can directly update its weights without needing any intermediate variables.

Practically, when a batch of $|\mathcal{B}|$ examples is fed to the MFB-DNN, the gradient update (42) will be approximately adjusted into the following form:

$$\begin{aligned}\widetilde{\boldsymbol{\theta}}^{[t+1]} = \widetilde{\boldsymbol{\theta}}^{[t]} - \frac{\eta}{|\mathcal{B}|} \sum_{n=1}^{|\mathcal{B}|} \Bigg\{ \Bigg[& \lambda_1 \nabla_{\widetilde{\varphi}_n} \mathcal{F}\left(\widetilde{\varphi}_n\right) \Big|_{\widetilde{\varphi}_n=\widetilde{\varphi}_n^{[t]}} \\ & + ((1-\lambda_1)/\lambda_2) \\ & \times \nabla_{\widetilde{\varphi}_n} \mathcal{M}\left(\left[\mathcal{C}\left(\widetilde{\varphi}_n\right)\right]^{+} \right) \Bigg|_{\widetilde{\varphi}_n=\widetilde{\varphi}_n^{[t]}} \Bigg] \\ \times & \left[\nabla_{\widetilde{\boldsymbol{\theta}}} \widetilde{\varphi}_n \Big|_{\widetilde{\boldsymbol{\theta}}=\widetilde{\boldsymbol{\theta}}^{[t]}} \right] \Bigg\}. \end{aligned} \tag{43}$$

Remark 2. *Although we only optimize the real-valued variable $\varphi \in \mathbb{R}$ in this paper, it is worth mentioning that our proposed approach may also be extended to the optimization of complex-valued variables. This may be achieved by doubling the number of output neurons to handle the real and imaginary parts. Another way of optimizing complex-valued variables is to use complex-valued DNNs [34]; however, this direction has not yet been fully supported by contemporary simulation tools.*

V. Numerical Results and Discussions

In this section, we will present our numerical results obtained by the proposed DNN. For characterizing the security vs reliability trade-off in terms of the SEPs vs the DCMC capacities, and the so-called intercept vs outage probability using the 4-QAM signalling. Note that we consider the uncoded transmission and leave coded schemes for future works.

Moreover, we consider the full search as the baseline since, it represents the standard best-case upper-bound. Hence, we compare both proposed DNN-based approaches to the full search for quantifying their security performance.

DNN configuration: The Lagrange-DNN and the MFB-DNN share the same configurations: the input, the output, the number of nodes (or neurons) in each layer, and the activation function of each neuron. The n-th input sample can be formulated as

$$\mathbf{x}_n = \left[\Re\left\{\mathbf{h}_{n,est}\right\}^{\top}, \Im\left\{\mathbf{h}_{n,est}\right\}^{\top} \right]^{\top},$$

where $\Re\{\cdot\}$ and $\Im\{\cdot\}$ denote the real and imaginary part, respectively. Herein, $\mathbf{h}_{n,est}$ is the realization of $\mathbf{h}_{est} = \left[\mathbf{h}_{\text{B},est}^{\top}, \mathbf{h}_{\text{E},est}^{\top}\right]^{\top}$ at the n-th time slot. As for the output $\mathbf{y}_{\text{out}}$, A simply designs $\mathbf{y}_{\text{out}} = \varphi$ in order to assign the optimal power allocation coefficient to the output node. Furthermore, φ assumes a real value in the range $(0, 1)$, thus the output layer can be activated by an activation function ranging from 0 to 1, for example, the sigmoid function. The number of hidden layers is 2 and the number of neurons in each hidden layer is 20. The learning rate of each DNN is set to 0.001. A batch of examples passed through a DNN is 2000, and the total number of examples is $20,000$. Unless otherwise specified, we will terminate the training process after 400 epoch.

A. Convergence

Figs. 3 and 4 characterize the convergence of both the Lagrangian and MFB solutions in comparison with the exhaustive search. The system parameters used are as follows: $\gamma_{\text{A}} = \{0, 5, \ldots, 35\}$, $N_{\text{A}} = 5$, $\epsilon = 0.7$ and $\zeta = 0$. Note that for the MFB approach, two cases are illustrated: i) $(\lambda_1, \lambda_2) = (0.5, 50)$ and ii) $(\lambda_1, \lambda_2) = (0.7, 10)$. Note that the number of evaluations used for the exhaustive search is equal to the number of examples used for the Lagrange/MFB-DNN. Observe that the Lagrangian performance is almost identical to that of the exhaustive search, while that of the MFB approach is different from the exhaustive search. In particular, P_{B} of the MFB-DNN is lower than P_{B} of the Lagrange-DNN, and similarly, P_{E} of the MFB-DNN is lower than P_{E} of the Lagrange-DNN. Naturally, we wish to reduce the value of P_{B} and simultaneously increase P_{E}, which is indeed the case in Fig. 3. This is because we are finding the optimal solution φ^* of the problem (24) in terms of the security vs reliability trade-off.

Similar to our observations gleaned from Figs. 3-4, the remaining figures also demonstrate a close agreement between the Lagrange-DNN and the exhaustive search, regarding the SEP, the outage probability and the intercept probability. By contrast, there is a significant gap between the MFB-DNN and the exhaustive search, although this gap is reduced for some specific parameter values. Noticeably, the MFB-DNN requires the careful tuning of the parameters λ_1 and λ_2 for approaching the exhaustive search performance, which has to be repeated when the system parameters change. By contrast, the Lagrange-DNN using Algorithm 1 does not require the tuning of any other parameters in the DNN architecture. Indeed, the Lagrange-DNN loss function does not rely on λ_1 and λ_2, but it relies only on the Lagrange multiplier ξ_{dual}, which will be automatically found by Algorithm 1.

It is clear that the Lagrange-DNN approaches the exhaustive search performance upon increasing its complexity, hence outperforming the MFB-DNN.

B. Security versus SNR Performance

Let us now discuss the security vs SNR performance in Figs. 3 and 4. In Fig. 3, P_{B} and P_{E} are depicted vs the logarithmic SNR scale in dB. Observe that P_{B} is always lower than P_{E} at any specific γ_{A}, because we find the optimal solution φ^* that minimizes P_{B} and maximizes P_{E} at the same time. At a high SNR of $\gamma_{\text{A}} = 20$ dB, P_{B} falls below 10^{-3}, while P_{E} can

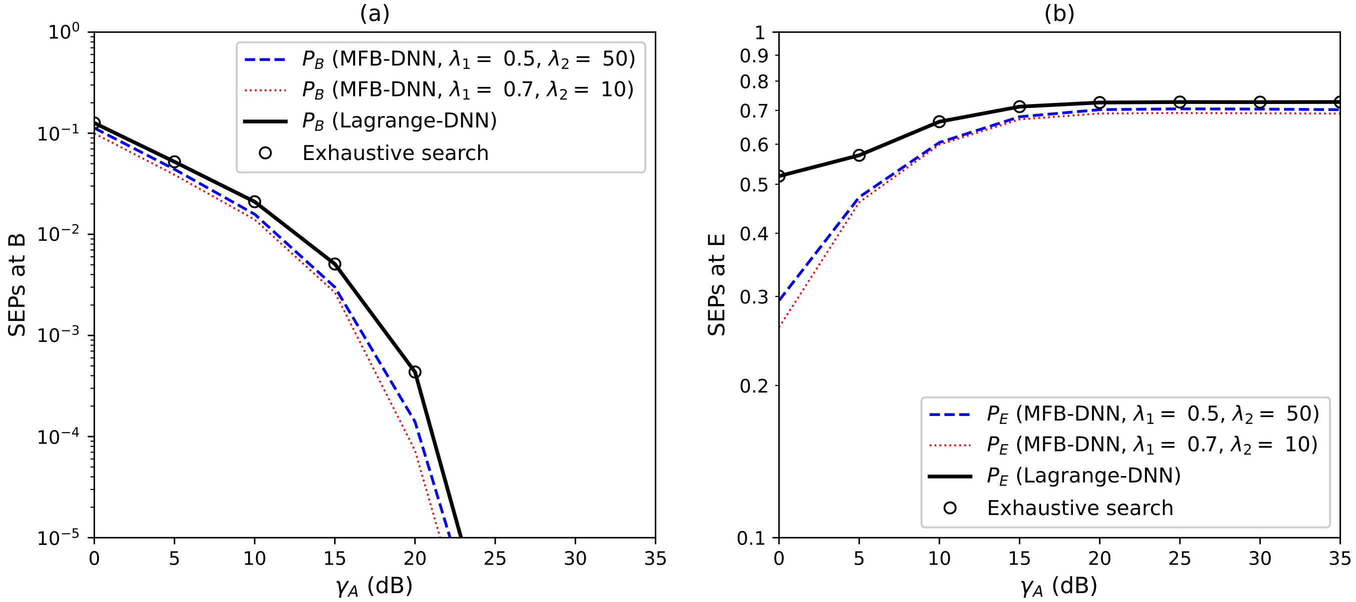


Fig. 3: Comparison between the Lagrange-DNN and MFB-DNN.

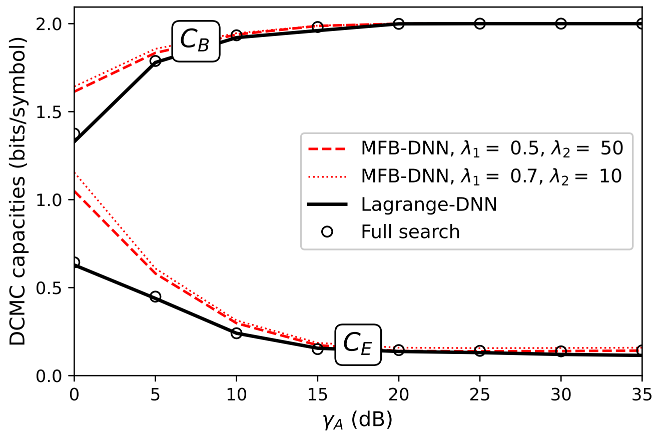


Fig. 4: C_{B} and C_{E} versus the SNR γ_{A}.

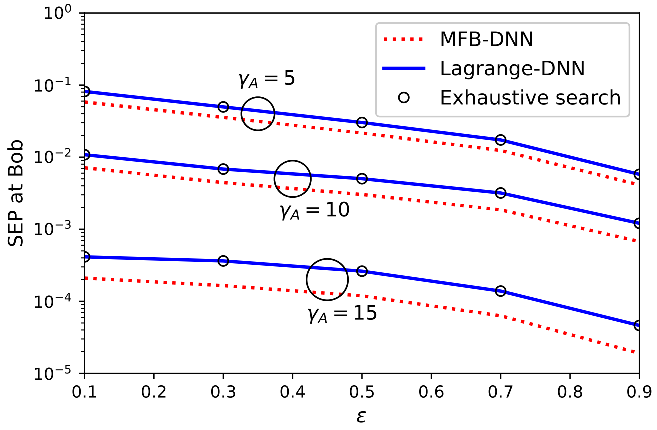


Fig. 5: SEP at $\mathtt{B}$ versus the priority factor ϵ.

exceeds 0.7. Moreover, P_{E} continues to increase with γ_{A}, but its increase tend to saturate around 0.72.

In Fig. 4, the average capacities of the legitimate and illegitimate channels are shown together. The trend of the legitimate channel capacity C_{B} is opposite to that of C_{E}. In particular, their gap widens as γ_{A} increases. As a result, the achievable DCMC secrecy rate, which is the difference between C_{B} and C_{E}, also increases with γ_{A}. At high SNR, C_{B} and C_{E} tend to a constant.

C. Security performance versus the priority factor

By choosing $\gamma_{\mathrm{A}} \in \{5, 10, 15\}$ dB, $N_{\mathrm{A}} = 15$, $\epsilon = 0.5$ and $\zeta = 0.3$, we will portray the effect of the priority factor ϵ on the SEPs. Figs. 5 shows P_{B} versus ϵ, while 6 shows P_{E} versus $\widehat{\epsilon} = 1 - \epsilon$. Recall that a large value of ϵ indicates a higher priority of minimizing P_{B}, while a larger of $\widehat{\epsilon}$ indicates a higher priority of maximizing P_{E}. This conflicting relationship between P_{B} and P_{E} is formulated in (23) and is illustrated by Figs. 5-6. In general, our objective is to decrease P_{B} and increase P_{E} at the same time. However, they are conflicting objectives, and there is a need to aim at a compromise.

Increasing ϵ leads to a reduction in P_{B}, because increasing ϵ implies giving higher priority to minimizing P_{B}. However, increasing ϵ also leads to a reduction of P_{E}, which then becomes beneficial for $\mathtt{E}$. To prevent $\mathtt{E}$ from successfully decoding the legitimate messages, we might want to increase the SEP at $\mathtt{E}$ by reducing ϵ, but the cost of reducing ϵ is an increased P_{B}. In short, it is impossible to have the lowest value of P_{B} and the highest value of P_{E} at the same time. Instead, we have to choose ϵ carefully in order to attain an acceptable pair of $(P_{\mathrm{B}}, P_{\mathrm{E}})$. For example, at $\gamma_{\mathrm{A}} = 5$ dB, if we want $P_{\mathrm{B}} \leq 10^{-2}$ and $P_{\mathrm{E}} \geq 0.6$ at the same time, then we can choose $\epsilon = 0.75$.

Still referring to Figs. 5 and 6, we can see that γ_{A} plays

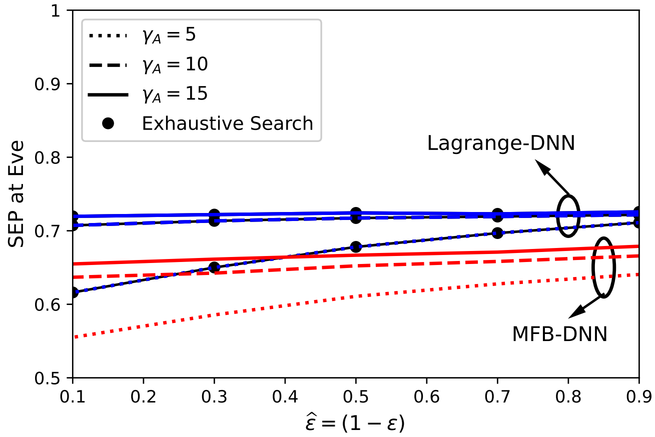


Fig. 6: SEP at $\mathbb{E}$ versus the priority factor $\widehat{\epsilon} = 1 - \epsilon$.

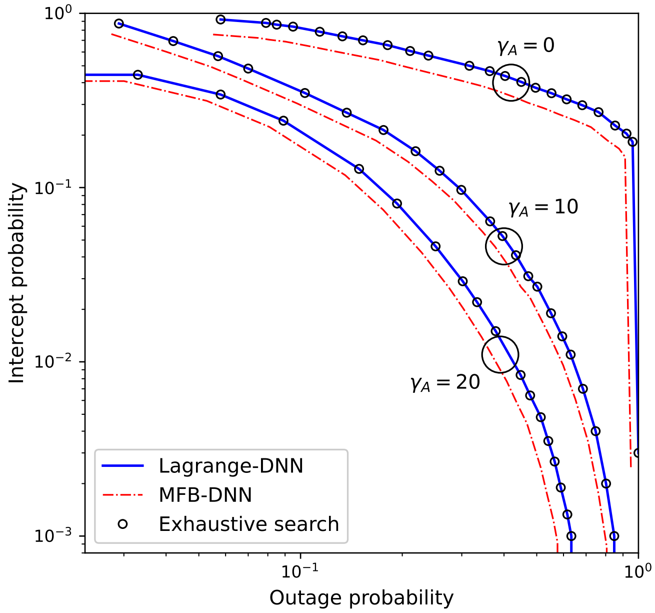


Fig. 7: IP versus OP with multiple values of $\gamma_{\rm A}$.

a more important role than ϵ in yielding an attractive performance. In particular, when $\gamma_{\rm A}$ increases from 5 dB to 15 dB, $P_{\rm B}$ reduces by three orders of magnitude, while $P_{\rm E}$ only increases slowly. For example, at $\gamma_{\rm A} = 10$ dB or 15 dB, the effect of ϵ on $P_{\rm B}$ is still significant, but it does not substantially affect $P_{\rm E}$. This implies that we should select ϵ as high as possible, when increasing $\gamma_{\rm A}$.

Although the SEP at $\mathbb{B}$ of Fig. 5 in the MFB-DNN approach is a little lower than that in the Lagrange-DNN approach, the SEP at $\mathbb{E}$ of Fig. 6 in the MFB-DNN approach is much lower than that in the Lagrange-DNN. Hence the MFB-DNN does not offer the optimal solution, because the OF constituted by the combination of $P_{\rm B}$ and $P_{\rm E}$ is not at its minimal value.

D. Intercept Probability versus Outage Probability

In order to further investigate the security vs reliability trade-off, it is worth contrasting the intercept probability (IP) with the outage probability (OP), where the IP is defined as $\Pr\{C_{\rm E} > R\}$ and the OP as $\Pr\{C_{\rm B} < R\}$. Here, $R \geq 0$ is the required data rate, which will be varied to measure the entire IP and OP range. Additionally, it is necessary to consider the worst-case PLS scenario of $C_{\rm E} \rightarrow C_{\rm E}^{\rm uppwer}$ and $C_{\rm B} \rightarrow C_{\rm B}^{\rm lower}$, hence we will portray the IP versus OP relationship in this scenario. Upon observing the security performance of the Lagrange-DNN approach and that of the MFB-DNN approach in terms of OP and IP, we find that the reduction of OP is attained at the cost of increasing IP and vice versa, regardless of the specific type of the DNN-based approach used.

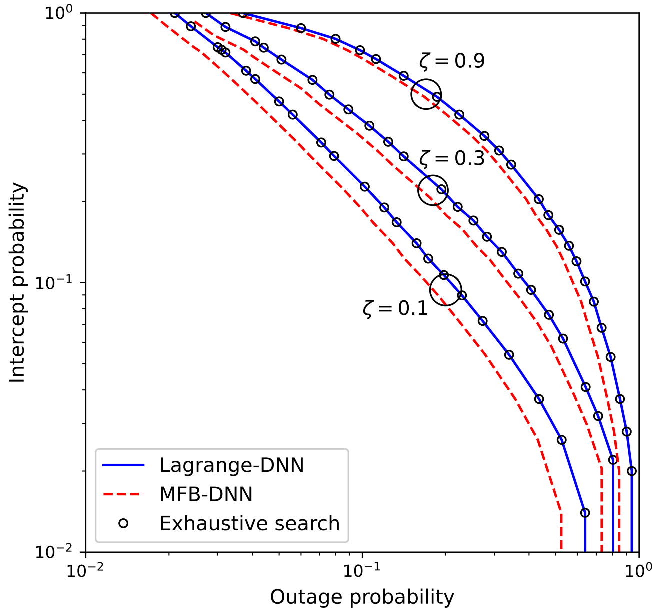


Fig. 8: IP versus OP with multiple values of ζ.

Explicitly, Fig. 7 portrays the IP vs OP relationship for different values of $\gamma_{\rm A}$, given that $N_{\rm A} = 15$, $\epsilon = 0.5$ and $\zeta = 0.3$, where as expected, the IP increases when the OP reduces and vice versa. This observation reflects the security (determined by the IP) vs reliability (determined by the OP) trade-off. More specifically, if we want to improve the security level by reducing the IP, we have to accept a lower level of reliability associated with an increased OP. Taking the SNR into account, we can see that a higher value of $\gamma_{\rm A}$ will improve the performance as a whole, i.e. both the IP and the OP are reduced, but their trade-off relationship still holds.

Fig. 8 also portrays the IP versus the OP, parameterized by the channel estimation error coefficient ζ, given that $\gamma_{\rm A} = 5$ dB, $N_{\rm A} = 5$, $\epsilon = 0.5$. The security vs reliability trade-off is, once again, confirmed by this figure, because we cannot reduce the IP and the OP at the same time. Instead, a lower IP can be attained upon accepting a higher OP. Furthermore, the performance is improved, when the estimation error coefficient ζ becomes smaller. For example, if the OP has to be below 10^{-1}, the IP is the lowest in conjunction with $\zeta = 0.1$. By contrast, the IP is the highest for $\zeta = 0.9$. The trends underline the importance of accurately estimating the channels.

VI. Conclusions

The security vs reliability trade-off was quantified in terms of the practical performance metrics of the SEPs and the DCMC capacity. We have shown that the optimization problem formulated can be solved both by the Lagrange-DNN and the MFB-DNN, but the former approaches the security performance of the exhaustive search more closely. Since the architecture is flexible enough to increase the number of output neurons for handling many other functional relationships, we will thus consider multivariate optimization problems in our future research. Arguably, spectrum sharing in 5G cognitive radio networks may pose a high risk of eavesdropping, because secondary users can exploit the spectrum sensing mechanism to violate the privacy of primary users. Since spectrum sensing can be supported by machine learning [35], it is necessary to design new machine learning-aided algorithms that can perform spectrum sensing and simultaneously guarantee the system as information security.

Appendix

A. Lower bound of C_B

We can rewrite (5) as follows:

$$r_{\mathrm{B}} = \sqrt{\varphi\gamma_{\mathrm{A}}}\,\mathbf{h}_{\mathrm{B}}^{\top}\mathbf{b}s_{\mathrm{info}} + \widehat{n_{\mathrm{B}}}, \tag{44}$$

where $\widehat{n_{\mathrm{B}}} = \sqrt{\frac{(1-\varphi)\gamma_{\mathrm{A}}}{N_{\mathrm{A}}-1}}\,\mathbf{h}_{\mathrm{B}}^{\top}\mathbf{A}\mathbf{z} + n_{\mathrm{B}}$ is the equivalent noise obeying a complex Gaussian distribution with zero-mean and variance $\widehat{\sigma_{\mathrm{B}}}^2 = \frac{(1-\varphi)\gamma_{\mathrm{A}}\|\mathbf{h}_{\mathrm{B}}^{\top}\mathbf{A}\|^2}{N_{\mathrm{A}}-1} + 1$, i.e., $\widehat{n_{\mathrm{B}}} \sim \mathcal{CN}\left(0, \widehat{\sigma_{\mathrm{B}}}^2\right)$.

If the entropy of $\widehat{n_{\mathrm{B}}}$ increases (i.e., equivalently, the noise variance $\widehat{\sigma_{\mathrm{B}}}^2$ increases), then the mutual information $I(s_{\mathrm{info}}; r_{\mathrm{B}})$ between s_{info} and r_{B} will reduce [36]. Since $\widehat{\sigma_{\mathrm{B}}}^2 \leq \frac{\gamma_{\mathrm{A}}\|\mathbf{h}_{\mathrm{B}}^{\top}\mathbf{A}\|^2}{N_{\mathrm{A}}-1} + 1 \triangleq \widetilde{\sigma_{\mathrm{B}}}^2$, we can have $I(s_{\mathrm{info}}; r_{\mathrm{B}}) \geq I(s_{\mathrm{info}}; \widetilde{r_{\mathrm{B}}})$ where

$$\widetilde{r_{\mathrm{B}}} = \sqrt{\varphi\gamma_{\mathrm{A}}}\,\mathbf{h}_{\mathrm{B}}^{\top}\mathbf{b}s_{\mathrm{info}} + \widetilde{n_{\mathrm{B}}} \tag{45}$$

and $\widetilde{n_{\mathrm{B}}} \sim \mathcal{CN}\left(0, \widetilde{\sigma_{\mathrm{B}}}^2\right)$. As a result, the DCMC capacity of the system formulated in (44), i.e., C_{B}, can be lower-bounded by that of the system in (45), i.e., $C_{\mathrm{B}}^{\mathrm{lower}}$. According to [23, Chapter 7], [24], the DCMC capacity $C_{\mathrm{B}}^{\mathrm{lower}}$ of the system in (45) can be calculated as shown in (16), which completes the proof. Note that the exact expression for C_{B} can be readily obtained by replacing $\widetilde{n_{\mathrm{B}}}$ and $\widetilde{\sigma_{\mathrm{B}}}^2$ in (16) by $\widehat{n_{\mathrm{B}}}$ and $\widehat{\sigma_{\mathrm{B}}}^2$.

B. Upper bound of C_E

The expression of C_{E} can be written as: [24]

$$\begin{aligned} C_{\mathrm{E}} &= \log_2 M - \frac{1}{M}\sum_{i=1}^{M}\mathbb{E}_{\mathbf{h},\mathbf{z},n_{\mathrm{E}}}\left\{\log_2\left[\sum_{j=1}^{M} e^{\Psi_{\mathrm{E}}(i,j)}\right]\right\} \\ &= \mathbb{E}_{\mathbf{h},\mathbf{z},n_{\mathrm{E}}}\left\{\log_2 M - \frac{1}{M}\sum_{i=1}^{M}\log_2\left[\sum_{j=1}^{M} e^{\Psi_{\mathrm{E}}(i,j)}\right]\right\}, \end{aligned} \tag{46}$$

where $\Psi_{\mathrm{E}}(i,j) = \left(-\left|\sqrt{\varphi\gamma_{\mathrm{A}}}\mathbf{h}_{\mathrm{E}}^{\top}\mathbf{b}(s_i - s_j) + \widehat{n_{\mathrm{E}}}\right|^2 + \left|\widehat{n_{\mathrm{E}}}\right|^2\right) / \widehat{\sigma_{\mathrm{E}}}^2$, $\widehat{n_{\mathrm{E}}} = \sqrt{\frac{(1-\varphi)\gamma_{\mathrm{A}}}{N_{\mathrm{A}}-1}}\,\mathbf{h}_{\mathrm{E}}^{\top}\mathbf{A}\mathbf{z} + n_{\mathrm{E}} \sim \mathcal{CN}\left(0, \widehat{\sigma_{\mathrm{E}}}^2\right)$, and $\widehat{\sigma_{\mathrm{E}}}^2 = \frac{(1-\varphi)\gamma_{\mathrm{A}}\|\mathbf{h}_{\mathrm{E}}^{\top}\mathbf{A}\|^2}{N_{\mathrm{A}}-1} + \frac{\sigma_{\mathrm{E}}^2}{\sigma_{\mathrm{B}}^2}$. The expectation $\mathbb{E}_{\mathbf{h},\mathbf{z},n_{\mathrm{E}}}\{\cdot\}$ is taken over $\mathbf{h}$, $\mathbf{z}$ and n_{E}.

The eavesdropper $\mathtt{E}$ is capable of improving its capacity by cancelling out n_{E} from (6). In this case, the ergodic DCMC capacity C_{E} of $\mathtt{E}$ can reach

$$C_{\mathrm{E}}^{\mathrm{upper}} = \lim_{\sigma_{\mathrm{E}}^2/\sigma_{\mathrm{B}}^2 \to 0} C_{\mathrm{E}}, \tag{47}$$

whose explicit form is shown in (18).

References

[1] Y. Wu, A. Khisti, C. Xiao, G. Caire, K. Wong, and X. Gao, "A survey of physical layer security techniques for 5G wireless networks and challenges ahead," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 4, pp. 679–695, 2018.

[2] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

[3] Z. Lin, M. Lin, W.-P. Zhu, J.-B. Wang, and J. Cheng, "Robust secure beamforming for wireless powered cognitive satellite-terrestrial networks," *IEEE Trans. on Cogn. Commun. and Netw.*, pp. 1–1, 2020.

[4] H. Fang, X. Wang, and S. Tomasin, "Machine learning for intelligent authentication in 5G and beyond wireless networks," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 55–61, 2019.

[5] S. N. Islam, Z. Baig, and S. Zeadally, "Physical layer security for the smart grid: Vulnerabilities, threats, and countermeasures," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6522–6530, 2019.

[6] L. Senigagliesi, M. Baldi, and E. Gambi, "Comparison of statistical and machine learning techniques for physical layer authentication," *IEEE Trans. on Info. Foren. and Sec.*, vol. 16, pp. 1506–1521, 2021.

[7] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, "Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks," *IEEE Access*, vol. 6, pp. 32 328–32 338, 2018.

[8] H. Huang, S. Guo, G. Gui, Z. Yang, J. Zhang, H. Sari, and F. Adachi, "Deep learning for physical-layer 5G wireless techniques: Opportunities, challenges and solutions," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 214–222, 2020.

[9] M. Eisen, C. Zhang, L. F. O. Chamon, D. D. Lee, and A. Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2775–2790, 2019.

[10] H. Lee, S. H. Lee, and T. Q. S. Quek, "Deep learning for distributed optimization: Applications to wireless resource management," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2251–2266, 2019.

[11] W. Lee, M. Kim, and D. Cho, "Transmit power control using deep neural network for underlay device-to-device communication," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 141–144, 2019.

[12] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning for the Gaussian wiretap channel," in *IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[13] D. He, C. Liu, H. Wang, and T. Q. S. Quek, "Learning-based wireless powered secure transmission," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 600–603, 2019.

[14] J. Xing, T. Lv, and X. Zhang, "Cooperative relay based on machine learning for enhancing physical layer security," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2019, pp. 1–6.

[15] S. Yun, J. Kang, I. Kim, and J. Ha, "Deep artificial noise: Deep learning-based precoding optimization for artificial noise scheme," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3465–3469, 2020.

[16] K. Besser, P. Lin, C. R. Janda, and E. A. Jorswieck, "Wiretap code design by neural network autoencoders," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3374–3386, 2020.

[17] J. Du, C. Jiang, H. Zhang, X. Wang, Y. Ren, and M. Debbah, "Secure satellite-terrestrial transmission over incumbent terrestrial networks via cooperative beamforming," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 7, pp. 1367–1382, 2018.

[18] J. Chen, L. Yang, and M. Alouini, "Physical layer security for cooperative NOMA systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4645–4649, 2018.

[19] J. Chen, Y. Liang, Y. Pei, and H. Guo, "Intelligent reflecting surface: A programmable wireless environment for physical layer security," *IEEE Access*, vol. 7, pp. 82 599–82 612, 2019.

[20] X. Chen, D. W. K. Ng, W. H. Gerstacker, and H. Chen, "A survey on multiple-antenna techniques for physical layer security," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1027–1053, 2017.

[21] B. Ji, Y. Li, S. Chen, C. Han, C. Li, and H. Wen, "Secrecy outage analysis of UAV assisted relay and antenna selection for cognitive network under Nakagami- m channel," *IEEE Trans. on Cogn. Commun. and Netw.*, vol. 6, no. 3, pp. 904–914, 2020.

[22] M. Maleki, K. Mohamed-Pour, and M. Soltanalian, "Receive spatial modulation in correlated massive mimo with partial csi," *IEEE Transactions on Signal Processing*, vol. 67, no. 5, pp. 1237–1250, 2019.

[23] L. Hanzo, O. Alamri, M. El-Hajjar, and N. Wu, *Near-capacity multi-functional MIMO systems: sphere-packing, iterative detection and cooperation*. John Wiley & Sons, 2009, vol. 4.

[24] Soon Xin Ng and L. Hanzo, "On the MIMO channel capacity of multidimensional signal sets," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 2, pp. 528–536, 2006.

[25] Y. Zou, B. Champagne, W. Zhu, and L. Hanzo, "Relay-selection improves the security-reliability trade-off in cognitive radio systems," *IEEE Transactions on Communications*, vol. 63, no. 1, pp. 215–228, 2015.

[26] T. Zheng and H. Wang, "Optimal power allocation for artificial noise under imperfect csi against spatially random eavesdroppers," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8812–8817, 2016.

[27] T. Yang, R. Zhang, X. Cheng, and L. Yang, "Secure massive mimo under imperfect csi: Performance analysis and channel prediction," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1610–1623, 2019.

[28] S. Timilsina, G. A. Aruma Baduge, and R. F. Schaefer, "Secure communication in spectrum-sharing massive MIMO systems with active eavesdropping," *IEEE Trans. on Cogn. Commun. and Netw.*, vol. 4, no. 2, pp. 390–405, 2018.

[29] A. Goldsmith, *Wireless communications*. Cambridge University Press, 2005.

[30] K. Hornik, M. Stinchcombe, H. White *et al.*, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[31] B. C. Csaji, "Approximation with artificial neural networks," *Faculty of Sciences, Eotvos Lorand University, Budapest, Hungary*, vol. 24, no. 48, p. 7, 2001.

[32] C. Sun, D. Liu, and C. Yang, "Model-free unsupervised learning for optimization problems with constraints," in *2019 25th Asia-Pacific Conference on Communications (APCC)*, 2019, pp. 392–397.

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

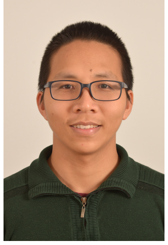
[34] S. Zhang, Y. Xia, and J. Wang, "A complex-valued projection neural network for constrained optimization of real functions in complex variables," *IEEE Trans. on Neural Netw. and Learning Sys.*, vol. 26, no. 12, pp. 3227–3238, 2015.

[35] X. Liu, Q. Sun, W. Lu, C. Wu, and H. Ding, "Big-data-based intelligent spectrum sensing for heterogeneous spectrum communications in 5G," *IEEE Wirel. Commun.*, vol. 27, no. 5, pp. 67–73, 2020.

[36] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

Tiep M. Hoang received the B.Eng. degree from the HCMC University of Technology, Vietnam, in 2012, the M.Eng. degree from Kyung Hee University, South Korea, in 2014, and the Ph.D. degree from the Queen's University of Belfast, UK, in 2019. He is currently a Research Fellow with the School of Electronics and Computer Science, the University of Southampton, UK. His current research interests include wireless security, flying ad-hoc networks, convex optimization, and machine learning.

Dong Liu [S'13, M'19] received his B.Eng. and Ph.D. degrees from Beihang University, Beijing, China, in 2013 and 2019, respectively. From 2019 to 2021, he was a research fellow at the University of Southampton, Southampton, U.K. He is currently an associate professor with the School of Cyber Science and Technology, Beihang University. His research interests include space-air-ground integrated networks, mobile AI, and mobile edge computing.



Thien Van Luong is a lecturer with the Faculty of Computer Science, Phenikaa University, Vietnam. He was a research fellow with University of Southampton, UK. Prior to that he did his PhD study at Queen's University Belfast, UK. He obtained his bachelor degree from Hanoi University of Science and Technology, Vietnam. His research interests include applied machine learning in signal processing and wireless communications.

Jiankang Zhang is a senior lecturer in Computer Science at Bournemouth University. Prior to joining in Bournemouth University, he was a senior research fellow at University of Southampton, UK. He serves as an Associate Editor for IEEE ACCESS.

Lajos Hanzo (FIEEE'04) received his Master degree and Doctorate in 1976 and 1983, respectively from the Technical University (TU) of Budapest. He was also awarded the Doctor of Sciences (DSc) degree by the University of Southampton (2004) and Honorary Doctorates by the TU of Budapest (2009) and by the University of Edinburgh (2015). He is a Foreign Member of the Hungarian Academy of Sciences and a former Editor-in-Chief of the IEEE Press. He has served several terms as Governor of both IEEE ComSoc and of VTS. He has published 2000+ contributions at IEEE Xplore, 19 Wiley-IEEE Press books and has helped the fast-track career of 123 PhD students. He is also a Fellow of the Royal Academy of Engineering (FREng), of the IET and of EURASIP. He is the recipient of the 2022 Eric Sumner Field Award.