IEEE Open Journal of Vehicular Technology

Received 27 August 2023; accepted 8 September 2023. Date of publication 13 September 2023;
date of current version 16 October 2023. The review of this article was coordinated by Editor Lina Bariah.

Digital Object Identifier 10.1109/OJVT.2023.3315216

Learning-Based Secret Key Generation in Relay Channels Under Adversarial Attacks

MEHDI LETAFATI [1] (Student Member, IEEE), HAMID BEHROOZI [1] (Member, IEEE), BABAK HOSSEIN KHALAJ [1,2] (Senior Member, IEEE), AND EDUARD A. JORSWIECK [3] (Fellow, IEEE)

[1]Department of Electrical Engineering, Sharif University of Technology, Tehran 11365-11155, Iran
[2]School of Computer Science, Institute for Research in Fundamental Sciences, Tehran 19395-5746, Iran
[3]Department of Information Theory and Communication Systems, Technische Universität Braunschweig, 38106 Braunschweig, Germany

CORRESPONDING AUTHOR: MEHDI LETAFATI (e-mail: mletafati@ee.sharif.edu)

Some preliminary results were presented in part at IEEE Global Communications Conference, Madrid, Spain, Dec. 2021 [1].

The work of Babak Hossein Khalaj was supported in part by the Institute for Research in Fundamental Sciences (IPM). The work of Eduard Jorswieck was supported in part by the Federal Ministry of Education and Research (BMBF, Germany) in the Program of "Souverän. Digital. Vernetzt." joint Project 6G-RIC, Project identification number: 16KISK031.

ABSTRACT Wireless secret key generation (WSKG) facilitates efficient key agreement protocols for securing the sixth generation (6G) wireless networks thanks to its inherently lightweight functionality. Nevertheless, with the existence of adversarial attacks or internal impairments, WSKG can be negatively affected during the randomness distillation, where the legitimate parties measure a source of common randomness. In this article, we propose a *learning-aided* approach for cooperative WSKG under man-in-the-middle (MitM) adversarial attack, while the legitimate nodes suffer from *hardware impairments* (HIs). The key idea is to process the PHY-attribute data on the application layer via deploying a deep neural network (DNN) to enhance the randomness distillation. This way, we realize a learning-based software-centric security solution. More specifically, we take into account the sequence-type nature of observed data, and propose a DNN comprised of *gated recurrent units* (GRUs) to learn the sequence of observations at legitimate endpoints, while the MitM is also alleviated. Our numerical results verify the performance gain of the proposed learning-based approach compared with the state-of-the-arts. Moreover, time and computation complexity of different learning-based models are studied to address the complexity-performance trade-off. Our tests highlight a performance gain of about 43% in terms of mean-square error (MSE) in comparison with a conventional PHY-only scheme.

INDEX TERMS Deep learning, man-in-the-middle adversarial attack, gated neural networks, cooperative key generation.

I. INTRODUCTION

Network security mechanisms rely traditionally upon cryptography-based keys to provide confidentiality and authentication requirements. Nevertheless, the modern era of the sixth generation (6G) wireless networks with substantially large number of peer-to-peer communications performed on-the-fly, challenges the performance of conventional solutions [2]. As a promising framework to realize a paradigm shift from the conventional complexity-based security solutions towards lightweight techniques, wireless secret key generation (WSKG) has been envisioned to be leveraged in future 6G networks [3], either as a standalone security mechanism or a complementary to the existing ones [4]. Notably, WSKG has gained much interest from both academic [4], [5], [6], [7] and industrial researchers [8], [9].

A. MOTIVATION

WSKG has remarkable merits for wireless networks. Specifically, its protocol does not require any additional infrastructure, and the secret key is obtained without the need of a third party contributor. This substantially reduces the required time for key agreement and the potential information leakage. Moreover, continuous update of the secret key

can be realized owing to the dynamic variations of wireless channels [6]. WSKG framework employs lightweight mechanisms with minimum required changes at the control plane (in terms of time scheduling, syncronization, and radio resource management), while offering information-theoretic security guarantees [10]. Hence, this approach is envisioned as a promising solution for the applications in beyond-the-fifth-generation (B5G) systems, such as the Internet-of-Things (IoT) [5], [7], [10] and latency-sensitive communications [3], [11]. Information-theoretic security guarantees of WSKG make this approach resilient against quantum computers, which can help the development of lightweight post-quantum security solutions [3]. In addition, the artificial intelligence and machine learning (AI/ML) techniques can be integrated into the WSKG scheme to improve its performance [3].

6G is envisioned to bring device-level intelligence via implementing contemporary deep learning (DL) algorithms [12]. Owing to the capabilities of DL methods to capture and learn from the feature statistics of data sequences, they can be incorporated into WSKG frameworks to realize intelligent security solutions [3], [13]. In addition, wireless networks are facing a new trend of transferring functionalities from PHY to higher layers via employing software-centric solutions. This key idea can be applied to the WSKG as well. That is, after obtaining the raw PHY-attribute data, it can be further exploited by DL algorithms implemented on higher layers of the protocol stack to improve system's performance [14].

The procedure of WSKG is based on exploiting the wireless medium as the shared source of randomness among legitimate entities. However, this phase of *randomness distillation* is affected by inevitable practical deficiencies such as random noise, imperfectly-reciprocal channel state information (CSI), and hardware impairments (HIs). In addition, substantial growth of adversarial attacks on wireless edge, such as spoofing and man-in-the-middle (MitM) attack—that are easily implemented using low-cost software defined radios—has been widely witnessed during recent years [14]. If not properly dealt with, such adversarial attacks have shown to be able to destruct the WSKG process [14], [15]. To elaborate, an adversary can try to "find" the agreed secret key between legitimate entities by inject poisoned signals, such that he "deceives" legitimate parties about the source of common randomness [15].

B. RELATED WORKS

To provide "proof of concept" (PoC) for WSKG, the authors in [7] implemented a WSKG scheme for long-range wireless communications in low power wide area networks (LPWANs). However, the effect of active or passive adversaries on the performance of their proposed scheme was missing. In [10], WSKG in the presence of both a totally-passive eavesdropper and a hostile jammer is analyzed, and a closed-form expression for the probability of successful handshaking is derived. The WSKG scheme was further extended to the case of cooperative communications in [16] and [17]. It is also of great importance to establish secure connections via wireless key agreement schemes, even when the endpoints lack a direct link to communicate. In such scenario, an intermediate node relays communication signals to the endpoints. In this regard, the authors in [17] investigated the key generation scheme in the presence of an intermediate relay node and an active jammer. The generated secret key was exploited to determine a secure frequency hopping pattern for IoT nodes. Cooperative secret key generation in static environments was investigated in [18] in the presence of a passive eavesdropper. To deal with poor sources of "natural randomness" in static environments, the authors proposed to induce "artificial randomness" to the network.

We remark that although the legitimate parties in [16], [17], [18] suffered from channel estimation errors, their transceivers were assumed to operate perfectly. That is, HIs were not modeled, nor examined in their proposed schemes. In addition, the active adversaries considered in the [10], [16], [17] were quite simple, i.e., they blindly inject noise-like signals to degrade the channel exploration capability of legitimate nodes. A MitM adversary who tries to control the WSKG process by spoofing the communication was investigated in [19]. The optimal strategy for the MitM in a direct communication was derived using game-theoretic approaches. However, the nodes were assumed to perform ideally (without any hardware mismatch) during transmission and reception. Then the authors in [20] considered the effect of mismatched radio-frequency (RF) front-ends in a point-to-point (P2P) communication, which sheds light on further investigations of the HI effects in cooperative WSKG scenarios.

Remarkably, none of the aforementioned works have leveraged the potential capabilities of ML/DL techniques to come up with an intelligent security solution for WSKG. In fact, to the best of our knowledge, there are only a few papers addressing learning-aided schemes for wireless key agreement protocols [21], [22], [23], [24]. For instance, [21] and [22] utilized a fully-connected (FC) neural network for a simple P2P communication in the presence of a simple passive eavesdropper. Considering a similar topology, convolutional neural networks were proposed in [23]. However, the sequence-type nature of observations during WSKG scheme was not addressed in [21], [22], [23]. In other words, they did not take into account the potential capabilities of state-aware neural networks, which can capture the relevant information within the chain of PHY data sequences. To the best of our knowledge, the scenario of learning-aided cooperative key generation is only studied in [24], however, the performance of that scheme under active adversarial attacks was not addressed. Moreover, we perform extensive evaluations on learning-based WSKG models and comprehensively compare our scheme with various benchmarks in terms of different secrecy and complexity metrics. Such extensive studies are not addressed in [24], nor in other related works in the context of cooperative key generation. Details of our novelties and contributions are explained in the following subsection.

C. OUR CONTRIBUTIONS

In this article, we propose a learning-aided solution for the cooperative WSKG scheme under MitM adversarial attack and the practical assumption of HIs. The MitM adversary in our system spoofs the randomness distillation phase via fake injection. The goal of a MitM is fundamentally different from that of an external hostile jammer (HJ) studied in [16] and [17]. HJ does not care about the procedure of key agreement, and its simple goal is to inject noise-like signals to impose mismatches within the observations of legitimate nodes. In contrast, a MitM performs his attack in a smart way that his injected adversarial data imitates a SoCR, and (if not secured) legitimate nodes can be misled about the SoCR. The corresponding mathematical details are given in Section III. To counteract the MitM, the exchanged packets at PHY are randomized, which is shown to prevent the MitM from taking control of the WSKG process. We further show in our article that our learning-based scheme is secure against the MitM adversarial attack in terms of having zero information leakage.

The main idea we pursue in this article is that we propose the exchanged data packets at PHY are subsequently processed on the application layer via implementing a *deep neural network* (DNN). This way, we come up with a data-driven and software-centric intelligent security solution for the cooperative WSKG scheme. Notably, *none of the reviewed literature considers the sequence-type context of data exchanged during WSKG protocol for capturing the relevant information within the chain of observation sequences.* On the contrary, we propose to employ state-aware neural networks. To elaborate, we leverage the concept of recurrent networks and propose a DNN comprised of *gated recurrent units* (GRU) to learn the sequence of observations at legitimate endpoints. We also compare our proposed learning-based scheme with different state-of-the-art neural networks to emphasize the performance of our scheme in terms of the resulting mean-square error (MSE) mismatch. Insightful comparisons from different aspects including time complexity (training time and inference time), computation complexity, and the required memory storage size are also provided in this article.

To have an accurate understanding of practical wireless systems, we also consider a realistic scenario, in which the transceivers of legitimate endpoints, as well as the intermediate relay, suffer from HIs, which was not addressed in [10], [11], [16], [17], [18], [19], [21]. Moreover, different from [7], [10], [16], [17], [19], we assume that the legitimate endpoints are faced with both channel estimation errors and imperfect reciprocity.

This article is an extension to our conference paper [1]. We extend the scheme of learning-based key generation to the case of *learning-based cooperative communication*, while in [1] we considered a simple scenario of Alice and Bob talking to each other directly. In this article, extensive simulations are also conducted to compare different learning algorithms in terms of various metrics, including MSE, training time, inference time, computation complexity, and storage usage, which were not addressed in [1] and [24]. We further emphasize that in [15], the performance analysis of a P2P WSKG scheme under HIs was provided (without proposing any learning algorithm to enhance the performance) to show that a fundamental limit occurs in terms of the achievable secret key rate. However, the goal of this article is totally different from [15], and we focus on proposing a *learning-based scheme* for cooperative WSKG under MitM attack, where we extensively study different learning algorithms and various benchmarks.

TABLE 1. Contrasting Our Proposed Scheme to the Existing Security Solutions at a Glance

	Our Scheme	[17]	[18]	[19]	[20]	[21]	[23]
HI Modeling	✓	✗	✗	✗	✓	✗	✗
MitM mitigation	✓	✗	✗	✓	✗	✗	✗
Learning-aided Solution	✓	✗	✗	✗	✗	✓	✓
Cooperative Communications	✓	✓	✓	✗	✗	✗	✗

To summarize, our contributions and novelties are as follows:

1) We propose a *learning-aided* approach for cooperative WSKG under MitM adversarial attack, considering a realistic scenario, in which the transceivers of legitimate endpoints and the intermediate node suffer from HIs. Outdated CSIs are also taken into account for the communication links.
2) Inspired by the concept of gated recurrent neural networks, we propose a DNN implementation to process the PHY-attribute data on the application layer. This way, the randomness distillation is enhanced and an intelligent learning-based security solution is realized. Studying the intersection of information theory and DL, we also show that implementing DNN does not leak any information about the key generation process.
3) The adversarial MitM attack against WSKG protocol is mitigated via employing randomized pilots (RPs). By leveraging an information-theoretic approach, we prove that the attack of MitM does not affect our WSKG scheme in terms of information leakage.
4) Our numerical studies shed more light on the effect of HIs and adversarial attacks on the DL-based WSKG. Moreover, insightful comparisons of our proposed approach with different benchmarks and other learning-based methods are provided in terms of MSE, secret key rate (SKR), key difference rate (KDR), number of sessions, computation time (training and inference), computation complexity, and memory storage size.

In Table 1, we provide a bold summary and explicitly contrast our contributions to the literature.

D. ORGANIZATION AND NOTATIONS

The rest of our article is organized as follows. We introduce a detailed description of our proposed system model in Section II. Our communication protocol and the attacking strategy of MitM adversary is addressed in Section III. Our implemented DNN is proposed in Section IV together with the technical details on different layers employed for it. Useful

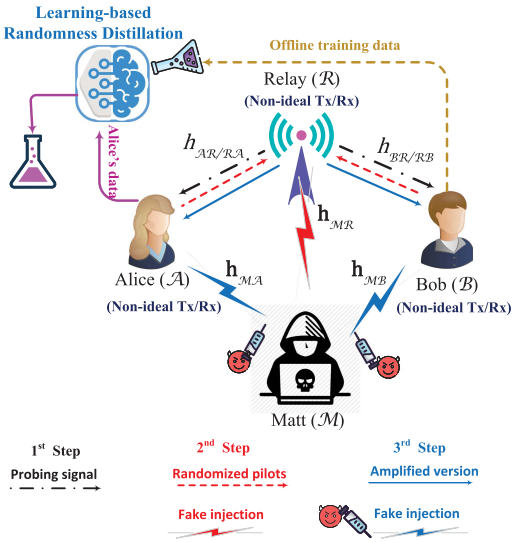

FIGURE 1. Proposed learning-assisted system model for hardware-impaired relay-aided WSKG under MitM attack.

information-theoretic analysis and remarks are provided in Section V to address the secrecy of our proposed scheme. Some information-theoretic aspects of utilizing a DNN for the WSKG framework are also addressed. Section VI provides readers with different tests and experiments on our proposed learning-aided scheme, and Section VII concludes the article.

Notations: We denote the transpose, conjugate, and ℓ^2 norm of a vector by $(\cdot)^{\mathsf{T}}$, $(\cdot)^{\dagger}$, and $||\cdot||$, respectively. Moreover, $|\cdot|$ represents the absolute value of a variable. The kernel (null) space is denoted by $\mathsf{null}(\cdot)$. Vectors are represented by bold lowercase letters, while matrices are written as bold uppercase symbols. The zero and the identity matrices are shown by $\mathbf{0}$ and $\mathbf{I}$, respectively. The real part of a variable x is illustrated by $\mathfrak{Re}(x)$. $\mathcal{CN}(\mu, \sigma^2)$ represents a complex Gaussian random variable (RV) with mean μ and variance σ^2. Moreover, the distribution of jointly Gaussian RVs X_1 and X_2 with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C} \geq 0$ is denoted by $(X_1, X_2) \sim \mathcal{CN}(\boldsymbol{\mu}, \mathbf{C})$. The expected value and the probability density function (pdf) of RV X are denoted by $\mathbb{E}[X]$ and $f_X(x)$, respectively. The mutual information of RVs X and Y is denoted by $I(X;Y)$. Hadamard (element-wise) product is denoted by $\odot$, while $\mathsf{sigm}(\cdot)$ and $\mathsf{tanh}(\cdot)$ stand for sigmoid and hyperbolic tangent functions, respectively.

II. SYSTEM MODEL

Our proposed system model consists of three mutually authenticated users, i.e., two legitimate endpoints, namely Alice ($\mathcal{A}$) and Bob ($\mathcal{B}$), and an intermediate relay ($\mathcal{R}$) node, as depicted in Fig. 1. Alice and Bob aim to agree on a common secret key sequence via exploiting the wireless medium, i.e, the communication links between $\mathcal{A}$-to-$\mathcal{R}$ and $\mathcal{B}$-to-$\mathcal{R}$. Notably, there does not exist a direct link between Alice and Bob due to heavy shadowing, or the direct link is too weak, such that $\mathcal{A}$ and $\mathcal{B}$ choose the option of relay-aided cooperative communication. This is a realistic assumption when the end nodes are placed far apart [17], [25]. Accordingly, an amplify-and-forward relay which is compliant with the networking protocols is employed, which amplifies and forwards its received signals without tampering with the contents. In our system model, there also exists a MitM adversary, named Matt ($\mathcal{M}$), who has impersonated other nodes and convinced them to establish unauthenticated links with him. This can be realized through circumventing mutual authentication mechanisms [28]. Investigation of authentication protocols and their vulnerabilities against MitM adversaries is out of the scope of this article. A DNN is implemented at $\mathcal{A}$ with the aim of compensating for observation mismatches among legitimate parties. This is done via learning Bob's observation sequence over time. To elaborate, the signals observed during packet exchange at PHY are processed on the application layer, through implementing a DNN to learn the sequence of observations. Then, the distilled randomness can be utilized as the source for key extraction. The model training and task inference procedures are implemented at one of the legitimate sides, and the intermediate node does not participate in the learning phase. Without loss of generality, it is assumed in the article that the DNN is employed by $\mathcal{A}$. In addition, we will show in Section V that if $\mathcal{M}$ is provided with the training data and similarly trains his DNN, he cannot obtain any useful information. Technical details on the structure and hyperparameters of the proposed DNN, together with the training strategy, are addressed in Section IV.

The physical RF transceivers of $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{R}$ suffer from HIs. The level of impairments at the transmitter and receiver hardware are denoted, respectively, by κ_n^{t} and κ_n^{r}, $n \in \{\mathcal{A}, \mathcal{B}, \mathcal{R}\}$. These factors reflect the error vector magnitudes (EVMs) as a measurable metric for the quality of RF transceivers [25]. While Alice and Bob exchange properly designed signals for randomness distillation, Matt tries to deceive legitimate entities by injecting fake signals. Details of the attacking model of Matt are elaborated in Section III. As a pessimistic assumption, Matt is considered to occupy an ideal RF transmitter to realize a powerful MitM attack, and he is equipped with $n_{\mathsf{T}} > 1$ transmitting antennas. We also assume that Alice, Bob, and the relay node are equipped with single antenna transceivers. This is in line with the scenario of low-cost devices employed in B5G IoT-enabled networks and with an advantage of the attacker [10], [17].[1]

Communication links are assumed to follow discrete time quasi-static block-fading model [10], [16], [17], [25]. Accordingly, the wireless link between Alice(Bob) and the relay is denoted by the complex circularly symmetric Gaussian RV $h_{\mathcal{AR}(\mathcal{BR})} \sim \mathcal{CN}(0, \delta^2_{\mathcal{AR}(\mathcal{BR})})$ where $\delta^2_{\mathcal{AR}(\mathcal{BR})}$ represents the large scale fading effect of legitimate channels. Similarly, the link of $\mathcal{R}$-to-$\mathcal{A}(\mathcal{B})$ is denoted by $h_{\mathcal{RA}(\mathcal{B})} \sim \mathcal{CN}(0, \delta^2_{\mathcal{A}(\mathcal{B})\mathcal{R}})$. As a practical assumption, we assume that the link of $\mathcal{R}$-to-$\mathcal{A}(\mathcal{B})$ experiences imperfect reciprocity with respect to the link of

[1] The proposed scheme can be directly applied to the generic case of multiple-input multiple-output (MIMO) scenario as well. The details are provided in the subsequent sections and through our numerical results.

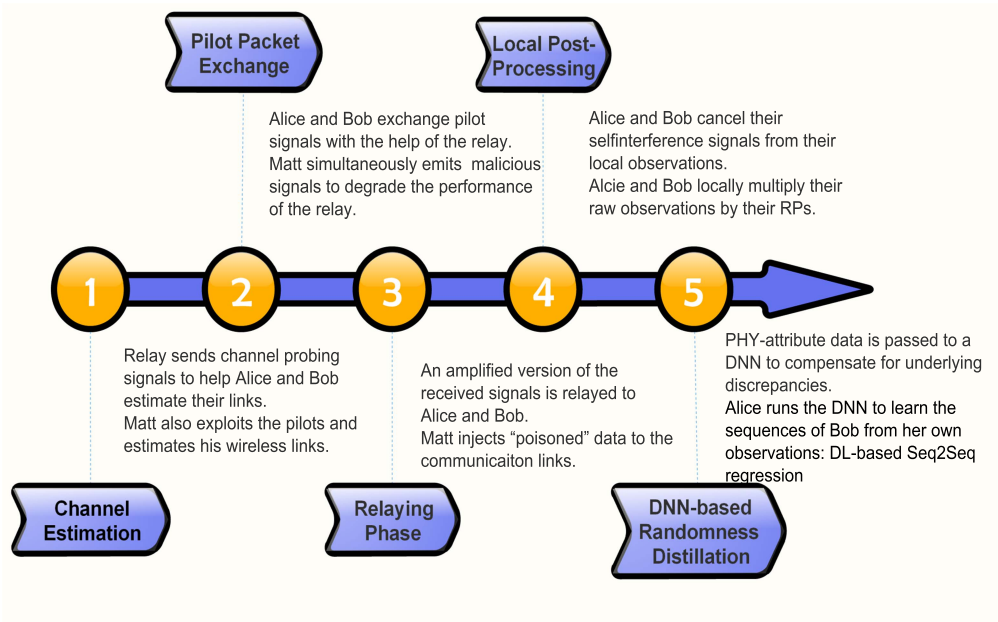

FIGURE 2. Block diagram of the proposed signaling and learning protocol.

$\mathcal{A}(\mathcal{B})$-to-$\mathcal{R}$ [9]. That is,

$$h_{\mathcal{R}\mathcal{N}} = \rho h_{\mathcal{N}\mathcal{R}} + \sqrt{1-\rho^2}u_{\mathcal{N}\mathcal{R}}, \quad \mathcal{N} \in \{\mathcal{A}, \mathcal{B}\} \tag{1}$$

where $h_{\mathcal{R}\mathcal{N}}$ and $h_{\mathcal{N}\mathcal{R}}$ have a correlation $0 < \rho \leq 1$, with $\rho = 1$ corresponding to the special case of perfect reciprocity. Moreover, $u_{\mathcal{N}\mathcal{R}}$ characterizes the uncertain part of $h_{\mathcal{R}\mathcal{N}}$ which is modeled as $u_{\mathcal{N}\mathcal{R}} \sim \mathcal{CN}(0, \delta^2_{\mathcal{N}\mathcal{R}})$ independent from $h_{\mathcal{N}\mathcal{R}}$. As a worst-case assumption from the security perspective [10], [17], [19], we consider that Matt has perfect knowledge about his channel vectors to Alice(Bob) and Relay, denoted by $\mathbf{h}_{\mathcal{M}A(\mathcal{B})}$ and $\mathbf{h}_{\mathcal{M}\mathcal{R}}$, respectively. Similar to [10], [17], [19], we assume channel coefficients of adversarial links are pairwise statistically independent with $\mathbf{h}_{\mathcal{M}n} \sim \mathcal{CN}(\mathbf{0}_{n_\mathsf{T}}, \delta^2_{\mathcal{M}}\mathbf{I}_{n_\mathsf{T}})$ for $n \in \{\mathcal{A}, \mathcal{B}, \mathcal{R}\}$, where $\delta^2_{\mathcal{M}}$ denotes the large scale effect. This is a plausible assumption in block-fading scenarios, while the channel's coherence time is respected [19]. Additive noises are assumed pairwise statistically independent with variance σ^2_n.

III. COMMUNICATION DESIGN

With the aim of distilling a common source of randomness from PHY, the legitimate nodes should first take turns conducting channel excursion, during which $\mathcal{A}$ and $\mathcal{B}$ exchange pilot signals with the help of the relay. Considering practical scenarios, physical RF transceivers suffer from impairments in real testbeds [26]. Hence, the general communication model from node i to node j at any given flat-fading channel is well-captured by the following equation [25], [26], [27].

$$y = h(s + \eta_{ij}) + n, \tag{2}$$

where $s \in \mathbb{C}$ with power $\mathbb{E}[|s|^2] = P_i$ is the signal sent over a wireless channel with fading coefficient $h \in \mathbb{C}$ and additive noise $n \in \mathbb{C}$, and y denotes the received signal. This is an experimentally-validated model for HIs, which is widely-adopted in wireless communication literature [25], [26].[2] The independent distortion noise $\eta_{ij} \sim \mathcal{CN}(0, \kappa^2_{ij}P_i)$ (varying from one block to another) models the HIs at the communication link of i-to-j, where $\kappa_{ij} \triangleq \sqrt{(\kappa_i^{\mathsf{t}})^2 + (\kappa_j^{\mathsf{r}})^2}$ reflects the *aggregate level of impairments*. $\kappa_i^{\mathsf{t}}, \kappa_j^{\mathsf{r}} \geq 0$ are the design parameters characterizing the level of impairments in the transmitter and receiver hardware, respectively [25], [26], [27].

In the following, we propose our communication protocol on how to distill a shared source of randomness for Alice and Bob using the characteristics of PHY layer. A general block diagram of the scheme is provided in Fig. 2.

A. SIGNALING PROTOCOL

As shown in Fig. 1, the legitimate nodes perform a three-step protocol to render randomness distillation from PHY, while Matt tries to deceive them via sending spoofing signals. The details of signaling protocols are as follows:[3]

[2]Detailed description of HIs and their compensation algorithms can be found in [27]. According to [25], the combined influence of different types of HIs at a given flat-fading block is well-modeled by the generalized channel model given in (2).

[3]Without loss of generality, the same protocol can be applied to the case of block-fading channels with N_{sc} parallel blocks, known as sub-channels [17], [18], [19]. In the following, we consider the communication over a single carrier for the sake of notation brevity and tractability.

1) ESTIMATION PHASE

In the first step, channel estimation is performed, during which the relay sends channel probing signals (with power $P_{\mathcal{R}}$) to help Alice and Bob estimate their links to $\mathcal{R}$. These estimates will further be utilized by $\mathcal{A}$ and $\mathcal{B}$ to cancel out self-interference signals from their observations. The estimates of Alice and Bob about their communication links to $\mathcal{R}$, denoted by $\hat{h}_{\mathcal{N}\mathcal{R}}, \mathcal{N} \in \{\mathcal{A}, \mathcal{B}\}$ are given as follows [17], [18]

$$\hat{h}_{\mathcal{N}\mathcal{R}} = h_{\mathcal{N}\mathcal{R}} + e_{\mathcal{N}}, e_{\mathcal{N}} \sim \mathcal{CN}\left(0, \delta^2_{\mathcal{N}\mathcal{R}}\kappa^2_{\mathcal{R}\mathcal{N}} + \frac{\sigma_n^2}{P_{\mathcal{R}}}\right), \quad (3)$$

where $e_{\mathcal{N}}$ denotes the estimation uncertainty, i.e., channel estimation error. According to (3), HIs at legitimate nodes can affect the quality of channel estimation. During the estimation phase, Matt also exploits transmitted pilots and obtains his link $\mathbf{h}_{\mathcal{M}\mathcal{R}}$ to the relay. Notably, the strategy of Matt is to perform his adversarial role in a "wait-then-attack" manner. He first listens to the exchanged packets to obtains an accurate estimation of his links to the other nodes. Once the CSI of his links to the legitimate nodes are estimated, he can design and transmit poisoned packets to inject fake SoCR at Alice and Bob. Otherwise, if he just blindly emits a noise-like signal from the very first step of packet exchange, he might be detected, while being unable to inject fake randomness.

2) PILOT PACKET EXCHANGE

Alice and Bob transmit pilot signals denoted by $x^{\mathsf{p}}_{\mathcal{A}}$ and $x^{\mathsf{p}}_{\mathcal{B}}$ with $\mathbb{E}[|x^{\mathsf{p}}_n|^2] \leq P_n, n \in \{\mathcal{A}, \mathcal{B}\}$, which is received by the relay $\mathcal{R}$, while Matt simultaneously emits his malicious signal x_j with power $\mathbb{E}[|x_j|^2] = P_{\mathcal{M}}$ towards $\mathcal{R}$ to degrade the receiving performance of $\mathcal{R}$. Hence, the received signal $y^{(2)}_{\mathcal{R}}$ at $\mathcal{R}$ can be formulated as

$$\begin{aligned} y^{(2)}_{\mathcal{R}} &= h_{\mathcal{A}\mathcal{R}}(x^{\mathsf{p}}_{\mathcal{A}} + \eta_{\mathcal{A}\mathcal{R}}) + h_{\mathcal{B}\mathcal{R}}(x^{\mathsf{p}}_{\mathcal{B}} + \eta_{\mathcal{B}\mathcal{R}}) \\ &\quad + \mathbf{h}^{\mathsf{T}}_{\mathcal{M}\mathcal{R}}\mathbf{p}^{(2)}_{\mathcal{M}}(x_j + \eta_{\mathcal{M}\mathcal{R}}) + n^{(2)}_{\mathcal{R}}, \end{aligned} \quad (4)$$

where the superscript (2) indicates the second step of pilot packet exchange. Moreover, $\mathbf{p}^{(2)}_{\mathcal{M}} \triangleq \frac{\mathbf{h}^{\dagger}_{\mathcal{M}\mathcal{R}}}{||\mathbf{h}_{\mathcal{M}\mathcal{R}}||}$ denotes the precoder of Matt. Details on how to choose the pilot signals $x^{\mathsf{p}}_{\mathcal{A}}, x^{\mathsf{p}}_{\mathcal{B}}$ are elaborated later.

3) RELAYING STEP

In the third step, an amplified version of $y^{(2)}_{\mathcal{R}}$ is relayed to Alice and Bob. The relaying gain G can be computed as $G = \sqrt{\frac{P_{\mathcal{R}}}{\mathbb{E}[|y^{(2)}_{\mathcal{R}}|^2]}}$ such that the mean transmit power of relay becomes $P_{\mathcal{R}}$ [29]. The value of G is determined based on the relay's received power $\mathbb{E}[|y^{(2)}_{\mathcal{R}}|^2]$, and it is considered as a publicly-known parameter [16], [17], [18].

Details of the Adversarial Attack: Now is the time for Matt to play his adversarial role. Informally speaking, the strategy of Matt is to "steal" the randomness distillation. That is, Matt aims to intelligently inject "poisoned" data so that the same fake signal is "observed" at legitimate endpoints, making them "believe" the source of shared randomness is what he has sent. Mathematically speaking, Matt injects an adversarially-precoded signal, denoted by $\boldsymbol{w}_{\mathcal{M}}$, such that his poisoned packets are observed similarly by Alice and Bob after they are received. Hence, Matt wants to satisfy

$$\mathbf{h}^{\mathsf{T}}_{\mathcal{M}\mathcal{A}}\boldsymbol{w}_{\mathcal{M}} = \mathbf{h}^{\mathsf{T}}_{\mathcal{M}\mathcal{B}}\boldsymbol{w}_{\mathcal{M}} \triangleq z_{\mathcal{M}}, \quad (5)$$

with $z_{\mathcal{M}}$ denoting the adversarial term observed by Alice and Bob. Inspired by (5), Matt designs his adversarial data, $\boldsymbol{w}_{\mathcal{M}}$, such that $(\mathbf{h}_{\mathcal{M}\mathcal{A}} - \mathbf{h}_{\mathcal{M}\mathcal{B}})^{\mathsf{T}}\boldsymbol{w}_{\mathcal{M}} = 0$. Hence, define the kernel (a.k.a. null-space) matrix $\mathbf{V} \in \mathbb{C}^{n_{\mathsf{T}} \times n_{\mathsf{T}}-1}$ associated with vector $\boldsymbol{v}^{\mathsf{T}} \triangleq (\mathbf{h}_{\mathcal{M}\mathcal{A}} - \mathbf{h}_{\mathcal{M}\mathcal{B}})^{\mathsf{T}} \in \mathbb{C}^{n_{\mathsf{T}}}$ as

$$\mathbf{V} \triangleq [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n_{\mathsf{T}}-1}] = \mathsf{null}(\boldsymbol{v}^{\mathsf{T}}). \quad (6)$$

Then, invoking (5) and (6), $\boldsymbol{w}_{\mathcal{M}}$ can be calculated as

$$\boldsymbol{w}_{\mathcal{M}} = \sum_{l=1}^{n_{\mathsf{T}}-1} \boldsymbol{v}_l x^{\mathcal{M}}_l, \quad ||\boldsymbol{v}_l|| = 1, \quad (7)$$

where $\boldsymbol{v}_l$ denotes the l'th column of $\mathbf{V}$ and $x^{\mathcal{M}}_l$ shows the adversarial signal on the l-th antenna (before precoding). Moreover, we have $\sum_{l=1}^{n_{\mathsf{T}}-1} \mathbb{E}[||\boldsymbol{v}_l x^{\mathcal{M}}_l||^2] \leq P_{\mathcal{M}}$, with $P_{\mathcal{M}}$ denoting Matt's transmit power budget. Remarkably, our proposed multi-stream MitM attack in (7) exploits the entire kernel space of attacking links, while setting $l = 1$ in (7) simplifies to the special case of single-stream injection proposed in [19]. We also remark that incorporating other types of learning-based adversarial attacks, such as adversarial machine learning, into the WSKG process will be studied in our future works.

Based on the aforementioned discussions and by utilizing (4)–(7), the raw observations of Alice and Bob, denoted by $\tilde{y}_{\mathcal{N}}$ for $\mathcal{N} \in \{\mathcal{A}, \mathcal{B}\}$ are as follows:

$$\tilde{y}^{(3)}_{\mathcal{N}} = h_{\mathcal{R}\mathcal{N}}(G y^{(2)}_{\mathcal{R}} + \eta_{\mathcal{R}\mathcal{N}}) + z_{\mathcal{M}} + \sum_{l=1}^{n_{\mathsf{T}}-1} \mathbf{h}^{T}_{\mathcal{M}\mathcal{N}}\boldsymbol{v}_l\eta_{\mathcal{M}\mathcal{N}} + n^{(3)}_{\mathcal{N}}. \quad (8)$$

Invoking (8), we can see that there exists a common (but adversarial) term $z_{\mathcal{M}}$ in the raw observations of Alice and Bob, which mimics the SoCR. However, it is the adversarial data sent by Matt, and hence, known by him. This can lead to security faults during the WSKG process, since both $\mathcal{A}$ and $\mathcal{B}$ maintain a common term which is known by Matt. Therefore, if we directly perform WSKG by exploiting the raw observations in (8), it results in information leakage to the MitM. Mathematically, the information leakage rate $\mathcal{L}$ for such a naive system is upper bounded by $\mathcal{L} \leq I(\tilde{y}_{\mathcal{A}}, \tilde{y}_{\mathcal{B}}; z_{\mathcal{M}})$.

Remark 1: Based on the aforementioned discussions, the optimal strategy for Matt (in the sense of maximizing the information leakage) is to choose his adversarial signals $\boldsymbol{w}_{\mathcal{M}}$ in a way that the observed signal at legitimate endpoints, $z_{\mathcal{M}}$, is a Gaussian-distributed RV. This fact basically relates to the capacity achieving input of Gaussian channels [17], [19], [42]. Therefore, by invoking the expression of $z_{\mathcal{M}}$ in (5) and (7), a good choice for Matt is to set $x^{\mathcal{M}}_l$ to a constant value.

This results in $z_{\mathcal{M}} \sim \mathcal{CN}(0, n_{\mathsf{T}} P_{\mathcal{M}} \delta^2_{\mathcal{M}})$, where the proof can be obtained in a similar way to what discussed in [19].

As a countermeasure to the MitM attacks, utilization of randomized pilots (RPs) have been shown to be an effective strategy [19], [30]. Hence, inspired by [19], we propose that in the pilot packet exchange, $\mathcal{A}$ and $\mathcal{B}$ exchange RPs of the form $\{\sqrt{P_n} e^{j\varphi_n}\}$, $n \in \{\mathcal{A}, \mathcal{B}\}$, where φ_n's are drawn according to independent and identically distributed (i.i.d.) zero-mean discrete uniform distribution $\{\pm\frac{\pi}{4}, \pm\frac{3\pi}{4}\}$, with $\mathbb{E}[x^{\mathsf{P}}_{\mathcal{A}}] = \mathbb{E}[x^{\mathsf{P}}_{\mathcal{B}}] = 0$, and $\mathbb{E}[x^{\mathsf{P}}_{\mathcal{A}} x^{\mathsf{P}}_{\mathcal{B}}] = 0$. We will show in Section V that this choice of employing RPs results in having zero information leakage to the MitM.

4) LOCAL PROCESSING

Alice and Bob cancel their self-interference signals from their local observations in (8). By invoking (1), (4), and (8), the self-interference signals at Alice and Bob can be formulated by $G\rho h^2_{\mathcal{AR}} x^{\mathsf{P}}_{\mathcal{A}}$ and $G\rho h^2_{\mathcal{BR}} x^{\mathsf{P}}_{\mathcal{B}}$, respectively. Since $\mathcal{A}$ and $\mathcal{B}$ have estimated versions of $\hat{h}_{\mathcal{AR}}, \hat{h}_{\mathcal{BR}}$, it results in $\hat{y}^{(3)}_{\mathcal{N}} = \tilde{y}^{(3)}_{\mathcal{N}} - G\rho \hat{h}^2_{\mathcal{AN}} x^{\mathsf{P}}_{\mathcal{N}}$, for $\mathcal{N} \in \{\mathcal{A}, \mathcal{B}\}$. After local interference cancellation, $\mathcal{A}$ and $\mathcal{B}$ locally multiply their raw observations $\hat{y}^{(3)}_{\mathcal{A}}$ and $\hat{y}^{(3)}_{\mathcal{B}}$ by their RPs to finally retain the source of shared randomness. This results in

$$y_{\mathcal{A}} = \hat{y}^{(3)}_{\mathcal{A}} x^{\mathsf{P}}_{\mathcal{A}} = \underbrace{\rho h_{\mathcal{AR}} h_{\mathcal{BR}} G x^{\mathsf{P}}_{\mathcal{B}} x^{\mathsf{P}}_{\mathcal{A}}}_{\text{Common Randomness}} + \underbrace{z_{\mathcal{M}} x^{\mathsf{P}}_{\mathcal{A}}}_{\text{Fake Randomness}} + \tau_{\mathcal{A}}, \tag{9}$$

$$y_{\mathcal{B}} = \hat{y}^{(3)}_{\mathcal{B}} x^{\mathsf{P}}_{\mathcal{B}} = \underbrace{\rho h_{\mathcal{AR}} h_{\mathcal{BR}} G x^{\mathsf{P}}_{\mathcal{A}} x^{\mathsf{P}}_{\mathcal{B}}}_{\text{Common Randomness}} + \underbrace{z_{\mathcal{M}} x^{\mathsf{P}}_{\mathcal{B}}}_{\text{Fake Randomness}} + \tau_{\mathcal{B}}, \tag{10}$$

$$\begin{aligned}\tau_{\mathcal{N}} &= \rho G \mathcal{E}_{\mathcal{N}} (x^{\mathsf{P}}_{\mathcal{N}})^2 + h_{\mathcal{RN}} G \\ &\times \left(h_{\mathcal{BR}} \eta_{\mathcal{BR}} + h_{\mathcal{AR}} \eta_{\mathcal{AR}} + v^{(2)}_{\mathcal{M}} + \mathbf{h}^{\mathsf{T}}_{\mathcal{MR}} \mathbf{p}^{(2)}_{\mathcal{M}} \eta_{\mathcal{MR}} + n^{(2)}_{\mathcal{R}} \right) x^{\mathsf{P}}_{\mathcal{N}} \\ &+ \sum_{l=1}^{n_{\mathsf{T}}-1} \mathbf{h}^{\mathsf{T}}_{\mathcal{MN}} \boldsymbol{\nu}_l \eta_{\mathcal{MN}} x^{\mathsf{P}}_{\mathcal{N}} + h_{\mathcal{RN}} \eta_{\mathcal{RN}} x^{\mathsf{P}}_{\mathcal{N}} \\ &+ \sqrt{1-\rho^2} u_{\mathcal{NR}} G \left(h_{\mathcal{AR}} x^{\mathsf{P}}_{\mathcal{A}} + h_{\mathcal{BR}} x^{\mathsf{P}}_{\mathcal{B}} \right) x^{\mathsf{P}}_{\mathcal{N}} + n^{(3)}_{\mathcal{N}} x^{\mathsf{P}}_{\mathcal{N}}. \end{aligned} \tag{11}$$

where $\tau_{\mathcal{A}}$ and $\tau_{\mathcal{B}}$ represent the residual HIs, channel estimation uncertainties, and random noises, given in (11), with $v^{(2)}_{\mathcal{M}} \triangleq \mathbf{h}^{\mathsf{T}}_{\mathcal{MR}} \mathbf{p}^{(2)}_{\mathcal{M}} x_j$ and $\mathcal{E}_{\mathcal{N}} \triangleq h^2_{\mathcal{NR}} - \hat{h}^2_{\mathcal{NR}}$ for $\mathcal{N} \in \{\mathcal{A}, \mathcal{B}\}$. Invoking (11), we can deduce that the relay has unintentionally amplified the components, such as HIs, which increases the level of mismatch within the signals of $\mathcal{A}$ and $\mathcal{B}$. This highlights the importance of proposing proper solutions for hardware-impaired cooperative key generation schemes as studied in this article. We also remark that if we set $\rho = 1$ and $\kappa^{\mathsf{t},\mathsf{r}}_{\mathcal{A},\mathcal{B}} = 0$, our network simplifies to the special case of ideally reciprocal channels and perfect hardware [19]. Moreover, if we set $P_{\mathcal{M}} = 0$, i.e., it results in the special case of relay-aided WSKG without adversarial attack [18]. In the following section, we propose our learning-based approach to enhance the hardware-impaired cooperative WSKG.

IV. NEURAL NETWORK IMPLEMENTATION

In the previous section, a general sketch of the shared randomness was achieved by performing proper packet exchange at PHY. Here, with the aim of improving the randomness distillation, the PHY data is passed to the application layer to be further processed and compensate for underlying discrepancies. Hence, a software-centric security solution is proposed by utilizing DNNs. Recall that the discrepancies exist due to the injected signals of MitM, and the unbalanced imperfections at legitimate transceivers. This can be inferred from (9) and (10). The main idea in this section is that we want to make predictions about the occupied data sequence of endpoints. By doing so, we wish to obtain a sequence similar to the original data occupied by either sides; hence, compensate for potential discrepancies between $\mathcal{A}$ and $\mathcal{B}$.

We leverage the concept of recurrent neural networks (RNNs) and capture the relevant information which lies within the chain of observation sequences. Remarkably, the chain-like nature of RNNs makes them suitable for sequence data types [32], [33]. RNNs allow information to persist, i.e., they do not begin to learn from scratch every time. Instead, at every time-stamp they learn from their previous understandings. There are feedback loops implemented in recurrent layers of RNNs to help them update their current state, according to the previous states. Thanks to the employment of feedback loops, the recurrent layers can memorize the historical information obtained from data sequences; hence, they are able to establish meaningful connections between every single data and its corresponding contextual information which is hidden in the data sequence [32], [33], [34].

A. OUR PROPOSED DNN

Inspired by the concept of RNNs, we propose a DNN for our WSKG scheme as depicted in Fig. 3. Our DNN is comprised of two GRU layers followed by two dense layers. According to Fig. 3, Alice runs a GRU-based DNN to learn the observation sequences of Bob from her own observations, $y_{\mathcal{A}}$. Technically speaking, our proposed neural network realizes a DL-based sequence-to-sequence (Seq2Seq) regression on Alice's observations to make them resemble Bob's.[4] Accordingly, the dense layers in our implemented DNN characterize the regression process on the underlying information of Alice's data, which is early extracted by the GRU layers. Remarkably, the GRU layers [35] as a well-established type of RNNs have become increasingly popular to be used in DL algorithms [36], [37]. GRUs maintain fewer tensor operations; hence, they typically perform faster than the long short-term memory (LSTM) networks during the training and inference.[5] Before going through the details of the learning and prediction

[4] Without loss of generality, the proposed neural network can be implemented at Bob by feeding $y_{\mathcal{B}}$'s to the network and predicting Alice's sequence. That is, we maintain the symmetry property in our proposed learning-based scheme.

[5] During our ablation studies, we evaluated both LSTMs and GRUs for our model and realized that almost the same performance could be achieved, while the GRU-based DNN performed faster than LSTM-based network.

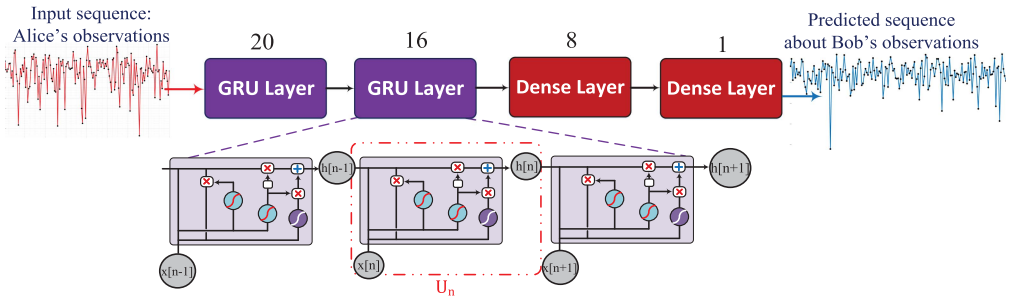

FIGURE 3. Proposed deep neural network implemented for the WSKG.

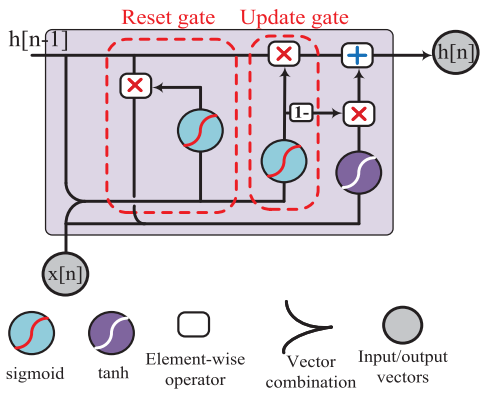

FIGURE 4. Detailed structure of a gated recurrent unit (GRU).

process, we briefly examine how GRU layers help networks extract state-aware information from given data sequences.

As illustrated in Fig. 3, a typical GRU layer consists of several units, called hidden units. The idea is to regulate the flow of information in a state-wise manner, i.e., the units maintain hidden states that act as the memory of neural networks, holding information on previous data the network has seen before. Hence, the GRU layer gradually learns which data in a sequence is important to keep or throw away. Then, by passing the relevant information down the chain of data sequence, it can perform predictions [36], [37]. After feeding each input data sequence to a GRU layer, it processes the input sequence one by one. During the processing of each element, the GRU layer passes the previous hidden state to the next states. To see how GRU layers calculate the hidden states, an arbitrary unit of a GRU layer is sketched in Fig. 4, showing the n'th hidden state. It is comprised of two main parts, i.e., a reset gate and an update gate. By defining the update and reset gate vectors as $\boldsymbol{z}[n]$ and $\boldsymbol{r}[n]$, respectively, and the output state vector as $\boldsymbol{h}[n]$, the controlling equations for a GRU are as follows:

$$\begin{aligned}
\boldsymbol{z}[n] &= \mathsf{sigm}\left(\mathbf{W}_z\boldsymbol{x}[n] + \mathbf{U}_z\boldsymbol{h}[n-1] + \boldsymbol{b}_z\right),\\
\boldsymbol{r}[n] &= \mathsf{sigm}\left(\mathbf{W}_r\boldsymbol{x}[n] + \mathbf{U}_r\boldsymbol{h}[n-1] + \boldsymbol{b}_r\right),\\
\tilde{\boldsymbol{h}}[n] &= \tanh\left(\mathbf{W}_h\boldsymbol{x}[n] + \mathbf{U}_h(\boldsymbol{r}[n]\odot\boldsymbol{h}[n-1]) + \boldsymbol{b}_h\right),\\
\boldsymbol{h}[n] &= \boldsymbol{z}[n]\odot\boldsymbol{h}[n-1] + (1-\boldsymbol{z}[n])\odot\tilde{\boldsymbol{h}}[n], \qquad (12)
\end{aligned}$$

where $\boldsymbol{x}[n]$ stands for the input vector to this unit (as shown in Fig. 4), $\{\mathbf{W}_z, \mathbf{W}_r, \mathbf{U}_z, \mathbf{U}_r\}$ and $\{\boldsymbol{b}_z, \boldsymbol{b}_r\}$ are the weights matrices and bias vectors, respectively, and $\tilde{\boldsymbol{h}}[n]$ formulates the intermediate memory unit (a.k.a. candidate state). According to (12), the update gate determines how useful past information is for the current state. The sigmoid function exploited in (12) leads to having updated values between 0 and 1. By invoking (12), the closer $\boldsymbol{z}[n]$ is to 1, the more we incorporate past information and vice versa. Reset gate helps the network ignore past information that might be irrelevant in future steps. Finally, the new candidate value $\tilde{\boldsymbol{h}}[n]$ is scaled by the GRU state update, and $\boldsymbol{h}[n]$ is calculated as the output. In the following, we study the utilization of GRU-based DNNs for the application of WSKG in our scheme.

B. TRAINING PROCEDURE OF THE PROPOSED DNN

During the training phase, the weights $\mathbf{W} \triangleq \{\mathbf{W}_z, \mathbf{W}_r, \mathbf{U}_z, \mathbf{U}_r\}$ and biases $\mathbf{B} \triangleq \{\boldsymbol{b}_z, \boldsymbol{b}_r\}$ in (12) need to be properly adjusted. This adjustment is done through training our DNN with a training set denoted by $\mathcal{T} \triangleq \{(\boldsymbol{y}_{\mathcal{A}}, \boldsymbol{y}_{\mathcal{B}})_i\}$, $i = 1, \ldots, \mathsf{N}_{\mathcal{T}}$, with $\mathsf{N}_{\mathcal{T}}$ showing the number of training sets. Moreover, $(\boldsymbol{y}_{\mathcal{A}}, \boldsymbol{y}_{\mathcal{B}})$ denotes the vector of occupied observations at Alice and Bob, each of which being a sequence of length L, where each element is given in (9) and (10), respectively. Using the examples provided in $\mathcal{T}$, our DNN gradually learns to predict the sequence of Bob. Mathematically speaking, the training process opt for adjusting the weights and biases of the DNN with the goal of minimizing the loss between desired output vector $\boldsymbol{y}_{\mathcal{B}}$ and the actual output sequence $\hat{\boldsymbol{y}}_{\mathcal{B}} = \mathcal{F}_{\mathbf{W},\mathbf{B}}(\boldsymbol{y}_{\mathcal{A}})$.

The training process should also take the information leakage into account. This can be captured by the mutual information metric between the adversarial signals occupied by Matt, and the data sequences at legitimate parties, i.e., $\mathcal{F}_{\mathbf{W},\mathbf{B}}(y_{\mathcal{A}})$ and $y_{\mathcal{B}}$. Hence, the overall loss function for the training process can be formulated as follows

$$\begin{aligned}
\{\mathbf{W}^*, \mathbf{B}^*\} = \underset{\mathbf{W},\mathbf{B}}{\operatorname{argmin}} \; & \frac{1}{\mathsf{N}_{\mathcal{T}}}\sum_{i=1}^{\mathsf{N}_{\mathcal{T}}} \ell\left(\mathcal{F}_{\mathbf{W},\mathbf{B}}(\boldsymbol{y}_{\mathcal{A}}), \boldsymbol{y}_{\mathcal{B}}\right)_i\\
& + I\left(\mathcal{F}_{\mathbf{W},\mathbf{B}}(\boldsymbol{y}_{\mathcal{A}})_i, \boldsymbol{y}_{\mathcal{B},i}; z_{\mathcal{M},i}\right), \qquad (13)
\end{aligned}$$

where $\ell(\cdot,\cdot)_i$ is any desired error measure between the input and output sequence of the DNN corresponding to the i-th training sequence. In this article, we employ mean-squared-error (MSE) $||\mathcal{F}_{\mathbf{W},\mathbf{B}}(\boldsymbol{y}_{\mathcal{A}}) - \boldsymbol{y}_{\mathcal{B}}||^2$ as a widely-adopted error measure. We note that in the next section, we show that the leakage term for the proposed scheme is zero. Hence, the final loss function will only consider the error measure between the input and output sequences. Invoking (9), (10), and (13), one can infer that the formulated optimization problem in (13) is complicated due to the existing non-linearities and fake signals. Hence, traditional optimization methods incur a considerable computational complexity. Whilst, finding the output of our DNN simply requires the calculation of learning blocks by moving from the input layer to the output layer of the trained DNN [13]. The minimization of (13) can be handled by off-the-shelf gradient descent-based methods specifically developed for training DNNs [38], where the review of these methods is beyond the scope of this article. We have chosen the widely-adopted adaptive moment estimation (Adam) optimizer algorithm. More details regarding the hyperparameters of our DNN, together with conducted experiments on our proposed network are provided in the subsequent section.

To prepare the training dataset $\mathcal{T}$, in addition to gathering $\mathsf{N}_{\mathcal{T}}$ observation sequences $\{\boldsymbol{y}_{\mathcal{A}}\}_i$, Alice should be provided with Bob's sequences $\{\boldsymbol{y}_{\mathcal{B}}\}_i$. This can be done via employing secure data transmission schemes for cooperative networks, e.g., the data transmission protocol proposed in [17]. It should be noted that sending the training set to Alice is for the purpose of training; not for quantization and key extraction. Hence, it will not compromise the secrecy. This is because the wireless channels change over time, and the observations which will be exploited to generate keys are independent from the ones used for training. In addition, we show in the next section that if Matt is provided with the training data $\mathcal{T}$ and implements the same DNN, he cannot obtain any useful information.

After the training process is completed, i.e., the minimization problem of (13) converges to a relatively low MSE, all weights and biases of our DNN are configured and the DNN achieves an acceptable state to perform Seq2Seq prediction. Once Alice and Bob perform packet exchange to distill PHY randomness, Alice will pass her new data sequences $\boldsymbol{y}_{\mathcal{A}}$ to the application layer to conduct DL-based prediction on Bob's data in a real-time manner. When the trained DNN is utilized for predicting new sequences, it is simply required to perform a forward propagation, i.e., moving forward through the DNN from the input layer to the output layer and performing the computations of (12). We note that the complexity of our DNN compared with related benchmarks is investigated in Section VI by examining the computational complexity, computation time, and memory size. Moreover, we show in Section VI that by considering different configurations and generating samples with different distributions than that of the training set, our DL-based approach performs well without the need to update the DNN.

V. SECURITY ANALYSIS AND DISCUSSIONS

In this section, we provide the information leakage analysis to address the security of our proposed learning-based scheme. Specifically, we show that the poisoned data of Matt (generated based on (5)–(7)) does not help him take control of the WSKG process, and the information leakage rate of our scheme is zero. We also address the intersection of information theory and deep learning, and show that utilizing the proposed DNN does not affect the information leakage rate. More precisely, if Matt is provided with the training data $\mathcal{T}$ and implements the same DNN, he cannot obtain any information corresponding to the WSKG process. To this end, we first show that the two endpoints experience independent versions of fake randomness in (9) and (10). The following corollary, which is obtained with a similar approach to what proposed in [1], formulates this claim.

Corollary 1: The fake randomness which lies within the observations of legitimate endpoints are pairwise independent with the following distribution

$$(z_{\mathcal{M}}x_{\mathcal{A}}^{\mathsf{P}}, z_{\mathcal{M}}x_{\mathcal{B}}^{\mathsf{P}}) \sim \mathcal{CN}\left(\mathbf{0}_{2\times 1}, P_{\mathcal{M}} n_{\mathsf{T}} \delta_{\mathcal{M}}^2 \begin{bmatrix} P_{\mathcal{A}} & 0 \\ 0 & P_{\mathcal{B}} \end{bmatrix}\right). \quad (14)$$

The corollary indicates that the adversarial counterparts lying within the observations of the legitimate parties do not have any mutual information with each other. In other words, $z_{\mathcal{M}}x_{\mathcal{A}}^{\mathsf{P}}$ and $z_{\mathcal{M}}x_{\mathcal{B}}^{\mathsf{P}}$ do not contain any common information. Hence, there is not any leakage imposed to the network through Matt.[6] According to the aforementioned discussions, by utilizing Corollary 1 and invoking (9) and (10), one can deduce that the information leakage $\mathcal{L}$ of the proposed scheme is zero due to the statistically independence of injected fake randomness at legitimate parties [1], [19]. Mathematically speaking, by invoking (9), (10), and (14) we can rewrite

$$\mathcal{L} \leq I(y_{\mathcal{A}}, y_{\mathcal{B}}; z_{\mathcal{M}}) = 0. \quad (15)$$

To gain insight about (15), it shows, from the information-theoretic perspective, that there will be no leakage by utilizing $y_{\mathcal{A}}$ and $y_{\mathcal{B}}$ for the process of secret key agreement, although Matt occupies the adversarial signal $z_{\mathcal{M}}$. (15) also ensures that the data sequences of Matt are decorrelated with the signals at Alice and Bob.

One might argue that according to the proposed scheme, Matt might obtain more information than $z_{\mathcal{M}}$ during the protocol, if equipped with a full-duplex radio, for example. To answer this, we provide the following remark.

Remark 2: Considering both an untrusted relay and an external pure eavesdropper, it is shown in [18] that the probability of eavesdropping attack can be arbitrary small. The results can be applied to our scenario when Matt is equipped with full-duplex radio, and wishes to simultaneously inject malicious data and wiretap the packet exchange. Therefore, inspecting the packets exchanged by Alice and Bob, e.g., by

[6] One can also easily verify that $z_{\mathcal{M}}$ and $v_{\mathcal{M}}^{(2)}$ are independent of the common randomness term, i.e., $\rho h_{\mathcal{AR}} h_{\mathcal{BR}} G x_{\mathcal{A}}^{\mathsf{P}} x_{\mathcal{B}}^{\mathsf{P}}$ in (9) and (10).

pretending to be the relay, does not help Matt obtain useful information. Similarly, listening to the packets transmitted by $\mathcal{R}$ does not help him in terms of the information leakage [18]. Also note that deploying full-duplex hardware and decoding all of the exchanged packets require consuming a relatively large amount of available energy, which is costly for an adversary. Hence, in this article we proposed a MitM who aims to wisely deceive endpoints regarding the SoCR via adding his own data to their observations.

Remark 3: Based on the above discussions, the achievable secret key rate (SKR) for the proposed scheme can be formulated by $R_{\mathsf{key}} = I(y_{\mathcal{A}}; y_{\mathcal{B}})$. However, obtaining a closed-form expression for the SKR in this case is intractable. This is because the common source of randomness in (9) and (10) corresponds to the product of two complex Gaussian RVs, i.e., $h_{\mathcal{AR}}x^{\mathsf{P}}_{\mathcal{A}}$ and $h_{\mathcal{BR}}x^{\mathsf{P}}_{\mathcal{B}}$.[7] This RV follows complex double Gaussian distribution, a.k.a. Gaussian-product, where its pdf is provided in [31].

The next question in terms of studying the secrecy of the proposed scheme is whether implementing the same DNN by Matt could help him infer any useful information. This is addressed in the following proposition.

Proposition: Leakage Analysis of Utilizing DNN: Intersection of Information Theory and DL. If Matt is provided with the training data $\mathcal{T}$ and implements the same DNN, denoted by $\mathcal{F}_{\mathbf{W},\mathbf{B}}(\cdot)$, he cannot obtain any useful information.

Proof: Based on the notations mentioned above, the inferred sequence at Matt, when utilizing the DNN, can be denoted by $\mathcal{F}_{\mathbf{W},\mathbf{B}}(z_{\mathcal{M}})$, which is obtained by feeding the adversarial samples $z_{\mathcal{M}}$ to the DNN. Accordingly, the leakage rate, $\mathcal{L}^{\mathsf{DNN}}$, in this case is bounded by the mutual information between the inferred sequence at Matt, and the data sequences at the output of the legitimate parties, i.e., $\mathcal{F}_{\mathbf{W},\mathbf{B}}(y_{\mathcal{A}})$ and $y_{\mathcal{B}}$. Mathematically speaking, we can write

$$\begin{aligned}\mathcal{L}^{\mathsf{DNN}} &\leq I\left(\mathcal{F}_{\mathbf{W},\mathbf{B}}(y_{\mathcal{A}}), y_{\mathcal{B}}; \mathcal{F}_{\mathbf{W},\mathbf{B}}(z_{\mathcal{M}})\right)\\ &\overset{(a)}{\leq} I\left(\mathcal{F}_{\mathbf{W},\mathbf{B}}(y_{\mathcal{A}}), y_{\mathcal{B}}; z_{\mathcal{M}}\right)\\ &\overset{(b)}{\leq} I\left(y_{\mathcal{A}}, y_{\mathcal{B}}; z_{\mathcal{M}}\right) \overset{(c)}{=} 0,\end{aligned} \tag{16}$$

where (a) follows from data processing inequality (DPI) [42] for the Markov chain $(\mathcal{F}_{\mathbf{W},\mathbf{B}}(y_{\mathcal{A}}), y_{\mathcal{B}})$—$z_{\mathcal{M}}$—$\mathcal{F}_{\mathbf{W},\mathbf{B}}(z_{\mathcal{M}})$. Similarly, (b) follows from DPI over $z_{\mathcal{M}}$—$(y_{\mathcal{A}}, y_{\mathcal{B}})$—$(\mathcal{F}_{\mathbf{W},\mathbf{B}}(y_{\mathcal{A}}), y_{\mathcal{B}})$. Finally, (c) directly follows from (15). Since the mutual information metric is non-negative, the leakage rate should be zero, and the proof is completed. We further note that invoking (16), the last inequality also indicates that utilizing the proposed DNN does not affect the information leakage rate.

According to the above proposition, one can deduce that $I(\mathcal{F}_{\mathbf{W},\mathbf{B}}(y_{\mathcal{A}}), y_{\mathcal{B}}; z_{\mathcal{M}}) = 0$. Therefore, invoking (13) and (16), we can rewrite the training process as follows

$$\{\mathbf{W}^*, \mathbf{B}^*\} = \underset{\mathbf{W},\mathbf{B}}{\operatorname{argmin}} \ \frac{1}{\mathsf{N}_{\mathcal{T}}} \sum_{i=1}^{\mathsf{N}_{\mathcal{T}}} \ell\left(\mathcal{F}_{\mathbf{W},\mathbf{B}}(\boldsymbol{y}_{\mathcal{A}}), \boldsymbol{y}_{\mathcal{B}}\right)_i. \tag{17}$$

Before providing the results of our numerical experiments, we mention that the full procedure of secret key agreement is realized through running the following blocks: 1) A mapping, e.g., quantization, from the occupied data of $\mathcal{A}$ and $\mathcal{B}$ to a discrete subspace, followed by 2) the reconciliation phase; and, 3) a hash function [18]. In this article, however, our focus is on the randomness distillation phase as the fundamental part of any WSKG scheme. Interested readers are referred to [4] for more details on other blocks of PHY-based key agreement. In our future works, we will study the integration of DL algorithms into the other blocks of WSKG.

TABLE 2. Parameters for Training the Proposed DNN

Learning Parameters	Values
Mini-batch size	100
Maximum training epochs	120
Initial learning rate	0.0005
Learning rate drop factor (every 20 epochs)	0.9
Number of training sequences	3500
Number of validation sequences	750
Training optimizer	Adam [39]

VI. NUMERICAL RESULTS

In this section, we present different numerical examples to investigate our proposed DL-based scheme for relay-aided WSKG. We also compare our scheme with different state-of-the-art benchmarks to demonstrate its performance. The codes are run on Intel(R) Xeon(R) Silver 4114 CPU running at 2.20 GHz. For the following tests, a typical wireless channel h between two arbitrary nodes with distance d is modeled as $h = Gd^{-\frac{\alpha}{2}}h_0$, with $G = \frac{c}{4\pi f_c}$ denoting the constant parameter of the path-loss with exponent $\alpha = 4$, $c = 3 \times 10^8$ m/s, and $f_c = 2.4$ GHz [1], [17], [18], [29], [40]. Moreover, $h_0 \sim \mathcal{CN}(0, 1)$ models the typical small scale Rayleigh fading. Unless otherwise stated, Alice, Bob and the relay are placed at $[-10, 0]$m, $[10, 0]$m, and $[0, 5]$m, respectively [18], [29], while Matt (equipped with $n_{\mathsf{T}} = 4$ antennas) is located at $[0, -5]$m [11], [17], [19]. This can be considered as a typical scenario of indoor WiFi networks. Training parameters are provided in Table 2. Moreover, the number of hidden units in GRU layers and the number of neurons employed at dense layers are denoted on top of their corresponding blocks in Fig. 3. In addition, dropout regularization with probabilities 0.8, 0.6, and 0.6 are implemented on each layer. The length of the input and output sequences of our DNN is set to $L = 20$, which is obtained by hyper-parameter tuning.[8] During the training of DNN, the transmit power of legitimate pilots and the MitM transmit power are set to $P_{\mathcal{A}\Leftrightarrow\mathcal{B}\Leftrightarrow\mathcal{R}} = 10$ dBm and $P_{\mathcal{M}} = 20$ dBm, respectively. The training set is created according to (9)

[7] To show that the mentioned RVs follow complex Gaussian distribution, one can easily follow a similar approach to [19].

[8] For the case of fast-varying channels instead of quasi-static block fading channels, input sequences of lengths less than 20 could be considered.

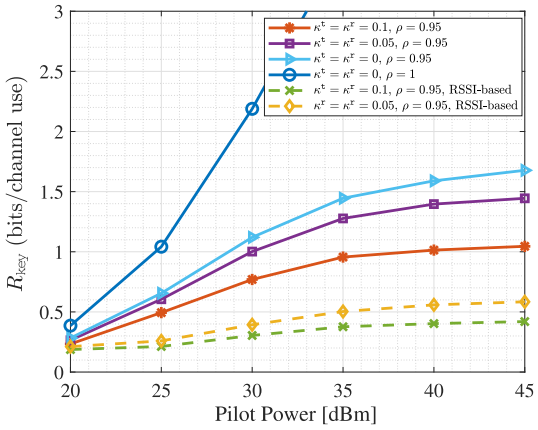

FIGURE 5. Secret key rate vs. pilot power for $P_{\mathcal{M}} = 10$ dBm with $\kappa_{\mathcal{A}}^{\mathrm{t(r)}} = \kappa_{\mathcal{B}}^{\mathrm{t(r)}} = \kappa_{\mathcal{R}}^{\mathrm{t(r)}} = \kappa^{\mathrm{t(r)}}$.

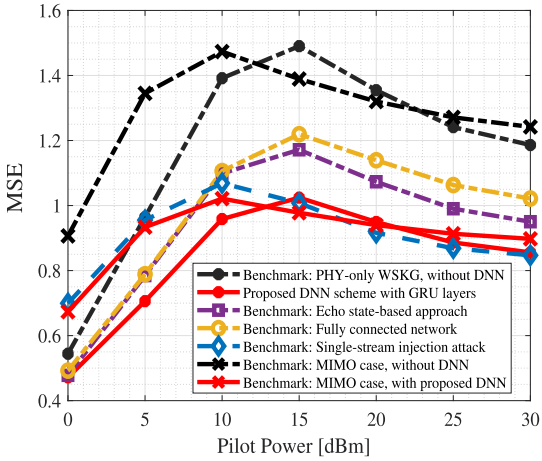

FIGURE 6. MSE between the sequences of Alice and Bob for $[\kappa_{\mathcal{A}}^{\mathrm{t}}, \kappa_{\mathcal{A}}^{\mathrm{r}}] = [0.12, 0.05]$, $[\kappa_{\mathcal{B}}^{\mathrm{t}}, \kappa_{\mathcal{B}}^{\mathrm{r}}] = [0.0875, 0.175]$, $[\kappa_{\mathcal{R}}^{\mathrm{t}}, \kappa_{\mathcal{R}}^{\mathrm{r}}] = [0.08, 0.15]$ [25], $\rho = 0.7$, and $P_{\mathcal{M}} = 10$ dBm.

and (10), and based on the configurations mentioned above, using Monte Carlo method. For the test scenario, however, we vary different configurations, such as transmit power, impairment levels, and nodes' locations, and generate samples with different distributions than the training set, in order to verify the generalization property of the implemented DNN.

Fig. 5 illustrates the achievable SKR R_{key} versus the transmit power of pilot packets for different HI levels. The mutual information calculation for the SKR is obtained numerically, using empirical distributions of $y_{\mathcal{A}}$ and $y_{\mathcal{B}}$ over 10^5 realizations for Monte-Carlo simulation [43], [44]. The figure demonstrates a fundamental limit of realistic WSKG schemes when HIs are taken into account. Considering wireless networks in practice, we face with the ceiling phenomena, i.e., the SKR saturation when increasing power. This ceiling effect can also be inferred from (2) and (9)–(11), where the increase in transmit power not only improves the quality of shared randomness, but also increases the variance of residual HI-related terms. The figure demonstrates that HIs are very influential at high SNR regimes, since the differences between SKR values at different HI levels are greater in high transmit powers. In addition, the figure indicates that the less HI the network faces, the more SKR can be achieved, which is in line with intuition. In this figure, we also examine some benchmarks: The SKR of WSKG scheme is plotted when the received signal strength indicator (RSSI) of $y_{\mathcal{A}}$ and $y_{\mathcal{B}}$ is considered as the source of randomness [4]. We observe that in this case, the achievable SKR is much lower than that of our scheme. This is because the RSSI-based scheme only utilizes the amplitudes of observations instead of the complex-valued observations $y_{\mathcal{A}}$ and $y_{\mathcal{B}}$. We also compare the hardware-impaired results with the special case of perfect hardware. We can infer from the figure (the line with triangle markers) that the imperfect reciprocity in wireless links also plays an important role in the ceiling effect. Moreover, when the HIs are neglected and the channel is assumed to be perfectly reciprocal—which is actually not realizable in a realistic deployment [9], a large gap occurs between the SKR of the ideal and the realistic scenarios. To conclude Fig. 5, it is pivotal for network designers to carefully take into account the hardware and channel imperfections to have an accurate understanding of wireless system.

Fig. 6 illustrates the observations mismatch, measured via (normalized) MSE metric, between the (absolute value of) occupied sequences at Bob and the predicted sequences at Alice. The MSE metric is plotted for different transmit powers $P_{\mathcal{A},\mathcal{B},\mathcal{R}} = P$. This figure provides a useful insight on choosing an appropriate transmit power for pilot packets. To elaborate, increasing P does not necessarily lead to achieving lower MSEs. In other words, if the signal level of the common randomness gets close to the received signal of fake data, the mismatch between legitimate parties increases according to (9) and (10). Fig. 6 also verifies that our proposed DNN is robust against different ranges of pilot power. In other words, our proposed DNN shows substantial reduction in observation mismatches for a wide range of transmit powers, although being trained by pilot packets with a fixed power $P_{\mathcal{A},\mathcal{B},\mathcal{R}} = 10$ dBm. Thus, we proposed a *data efficient* DNN, which does not need to be retrained when the pilot powers change.

In this experiment, we also investigate the performance of our proposed DNN compared with different state-of-the-arts We also show the generalization capability of our DL-based approach for being utilized in different communication scenarios. Notably, the following DL-based and non DL-based benchmarks are considered:

1) PHY-ONLY WSKG SCHEME

Fig. 6 shows that more than 40% improvement, in terms of distillation mismatch between Alice and Bob, is achieved by implementing our proposed DNN compared with a PHY-only WSKG scheme [16], [18], which only relies on PHY-extracted observations rather than employing a neural network.

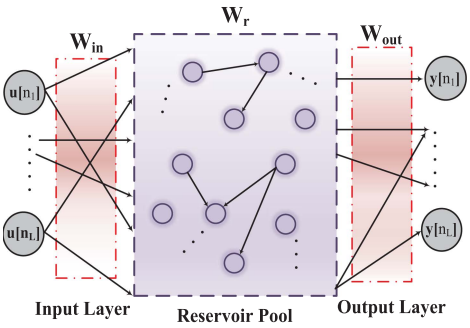

FIGURE 7. General sketch of the ESN benchmark.

2) ECHO-BASED NEURAL NETWORK

An echo state network (ESN) is implemented as a benchmark for predicting the observations sequences [41]. ESNs perform prediction using a relatively large reservoir of sparsely-connected neurons, each of which has a short-term memory of the previously-seen states. The main idea of ESNs is that the sparse random connections in the reservoir pool let previous states "echo" even when they have passed. After data echoes in the pool, it flows towards the output layer. The recurrent connections in the reservoir pool together with the connection weights in the input layer are randomly generated, while the output layer (which connects the reservoir to the output neurons) is trained during the training process. A general sketch of a typical ESN is illustrated in Fig. 7. For this benchmark, we implemented an ESN with a pool of size $N_r = 50$ neurons. We also set the spectral radius of the reservoir weights to 0.5 with connection density of $s_p = 0.5$. In addition, the weights of all untrained connections were chosen uniformly between -1 and 1. As can be seen from Fig. 6, our proposed GRU-based scheme outperforms the ESN benchmark by about 15%. This performance is achieved thanks to the wisely-adopted reset and update mechanism of GRU layers proposed in (12), while the typical update equations of neuron reservoir is a simple echo-inspired update (Please see [41] and [24] for the detailed mechanism of ESNs).

3) FULLY-CONNECTED (FC) NEURAL NETWORK

The FC network is implemented for another learning-based benchmark [21], [22]. For this benchmark, we implemented two dense layers with 8 neurons at hidden layer and 5 neurons at output layer. Remarkably, our proposed DNN is comprised of both the GRU layers and the dense layers. Therefore, in addition to having the learning capabilities of a FC network, our DNN is also capable of capturing the relevant information which lies within the sequence of observations. Hence, better performance can be achieved compared with a simple FC network by about 20% performance gain.

Comparing the general structure of our DNN, which is comprised of recurrent and dense layers (Figs. 3 and 4), the ESN benchmark, which is an aggregated version of recurrent neurons with a sparsely-connected network (Fig. 7), and a general FC network, one can intuitively imply that the prediction performance of an ESN would be something between the performance of a FC network and a GRU-based network, where our results in Fig. 6 validate this claim.

TABLE 3. Performance Comparison Between the Proposed DNN and Benchmarks

Metric	GRU	ESN	FC
Training time	120.37s	84.3ms	17.53s
Inference time	13.0ms	0.28ms	11.5ms
Memory Size	16kB	1315kB	2kB

4) FURTHER COMPARISONS BETWEEN GRU, ESN, AND FC

To have a more comprehensive insight on performance comparisons between the GRU-based neural network, the ESN, and the FC network, we further examined the required training time (with a fixed training data size), the inference time for predicting new sequences (during a fixed number of 256 time-stamps), and the required memory storage for saving each of the corresponding neural networks after that they are trained. These experimental results are summarized in Table 3. According to the table, the implemented ESN maintains a small training time compared with FC and GRU-based networks. This basically addresses the typical advantage of echo-based approach, i.e., its incredibly simple training process as the output layer is the only layer that gets trained, while other weights are randomly-assigned just once. To address the performance-complexity trade-off, we mention that the computation time of the ESN is less than the GRU-based approach and FC network, thanks to its relatively simple recurrent structure with sparse connections. However, our proposed DNN achieves much lower MSEs than the ESN and FC, as shown in Fig. 6. This can be interpreted as a trade-off between computation time/complexity and resultant MSE. Notably, the required memory storage to save the trained neural network is drastically large for the ESN. This is due to its huge number of internal states in the reservoir pool which needs to be stored for inference on new data.

Based on the results of Table 3, one might argue that the training time for our GRU-based network is too long. Although it seems to contradict with the purpose of utilizing our DNN, however, this is not the case due to the following reasons: Training is performed offline before establishing the real-time configuration settings. Hence, much higher computational time can be afforded with significantly less constraints than a real-time computation [13]. Once the offline training is finished, it can be used for online prediction of new data sequences in a real time manner, where our results show that better performance than ESN and FC networks can be achieved with much less time for online computations than the offline training.

We further study the computational complexity of our scheme and the state-of-the-art benchmarks. The computational complexity is evaluated in terms of the number of floating point operations (FLOPs) [45], as given in Table 4. In this table, l_i and l_o denote the length of input and output

TABLE 4. Computational Complexity of the Proposed DNN and Benchmarks

Neural Architecture	FLOPs
GRU	$\mathcal{O}\left(\sum_{i=1}^{H} n_{h_i}(n_{(i-1)} + n_{h_i} + 1)\right)$
ESN	$\mathcal{O}\left(N_r(l_i + N_r s_p + 2 + l_o)\right)$
FC	$\mathcal{O}\left(l_i n_1 + \sum_{i=1}^{H-1} n_i n_{i+1} + l_o n_H\right)$

vectors of the corresponding neural networks, (which is set to 20 in our numerical experiments). For the ESN benchmark, N_r and s_p denote the number of internal neurons within the reservoir pool, and the sparsity parameter, respectively. Finally, n_i and n_{h_i} $(1 \leq i \leq H)$ respectively stand for the number of neurons in the dense layers of FC benchmark, and the number of hidden states in the recurrent layers of our GRU-based scheme, with H denoting the number of neural layers according to Fig. 3. Inspecting the computational complexity orders in Table 4, one can imply that the computation complexity of the studied benchmarks are more or less the same, with the same polynomial order $\mathcal{O}(n^2)$ with respect to the size of the employed neurons. This is also in line with the inference computation time results in Table 3. Nevertheless, we emphasize that inference computation time in Table 3 is comprised of not only the tensor-based multiplications, but also other operations and processes, including additions, concatenations, activation functions, and reading from and writing to the memory, which would be different among different neural network architectures, and investigating their corresponding mechanisms is beyond the scope of this article.

We finally emphasize that moving from software level computations towards hardware level implementations, other computational complexities could be taken into account, such as number of bit operations (BOP), number of additions and bit shifts (NABs) in fixed-point computations, and number of hardware logic gates [45], which are not the focus of this article, as we proposed a software-centric security solution by employing a DNN at the application layer of the network protocol stack.

5) SINGLE-STREAM ATTACK

Based on the adversarial attack elaborated in (7), the MitM launches a multi-stream injection attack in our system. A special case of single-stream attack can also be considered by choosing an arbitrary column of (6) as a benchmark [19]. Although the training procedure has been performed under multi-stream attack, the result of our tests in Fig. 6 shows that our DNN is also robust in the single-stream scenario.

6) MULTI-ANTENNA WSKG

To highlight the generic capability of our proposed scheme, a MIMO WSKG scheme is considered in this benchmark, where Alice and Bob are equipped with 3 antennas. Remarkably, it can be seen from Fig. 6 that our DL-based approach can also be applied to the case of multi-antenna legitimate nodes by achieving 40% performance gain compared to a conventional PHY-only MIMO WSKG scheme [29].

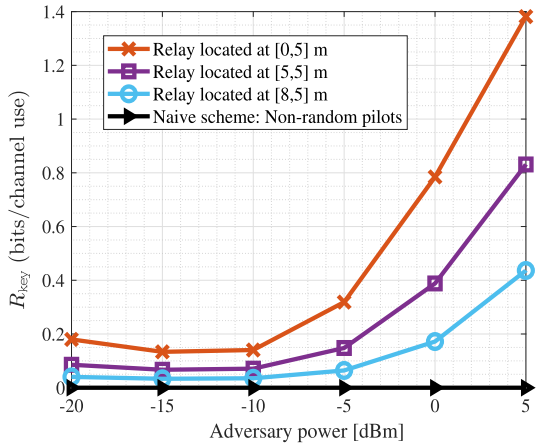

FIGURE 8. Achievable SKR vs. adversary power, for $\rho = 0.8$ and $[\kappa^{t}_{\mathcal{A},\mathcal{B},\mathcal{R}}, \kappa^{r}_{\mathcal{A},\mathcal{B},\mathcal{R}}] = [0.05, 0.05]$.

Remark 4: Trade-off between MSE and SKR upper bound. We emphasize that implementing DNNs cannot increase the "achievable" SKR, due to DPI. Mathematically, we have $I(\mathcal{F}_{\mathbf{W},\mathbf{B}}(y_{\mathcal{A}}); y_{\mathcal{B}}) \leq I(y_{\mathcal{A}}; y_{\mathcal{B}})$. However, as we can see from Fig. 6, the MSE of observations is decreased by using the proposed DNN. This can facilitate having lighter information reconciliation algorithms for error correction, resulting in less information leakage and communication overhead during the reconciliation phase. It can be an interesting research direction to investigate the trade-off between utilizing DNNs vs. employing reconciliation algorithms in the future works.

Fig. 8 shows the SKR versus the MitM adversarial power $P_{\mathcal{M}}$ when $\mathcal{R}$ is placed in different locations. Pilot packets with power 0 dBm is considered for this test. The figure highlights the fact that when the relay moves towards one of the legitimate parties, the achievable SKR decreases. This is because in a non-symmetric placement of legitimate nodes, higher levels of discrepancies between $y_{\mathcal{A}}$ and $y_{\mathcal{B}}$ occurs. This is because the aggregate levels of HIs in relaying links $h_{\mathcal{AR}}$ and $h_{\mathcal{BR}}$ are different. This can also be inferred from the residual terms in (11). In this figure, the SKR of a conventional scheme with unmodified pilot signals is also depicted, which shows that the MitM can override the key generation process if RPs are not exploited. This can be inferred from (8), in which a common adversarial data $z_{\mathcal{M}}$ (designed, controlled, and injected by Matt) lies within the observations of Alice and Bob. In other words, the information leakage in this case can become arbitrarily large with the increase in $P_{\mathcal{M}}$. However, thanks to the exploitation of RPs, Matt cannot disrupt our proposed scheme via increasing his power. Instead, the adversary should choose $P_{\mathcal{M}}$ in a way that the received signal level of his adversarial data $z_{\mathcal{M}}$ gets closer to the signal level of shared randomness data. By doing so, the mismatches between legitimate endpoints increases and the achievable SKR decreases. A similar trend was seen in Fig. 6 as well.

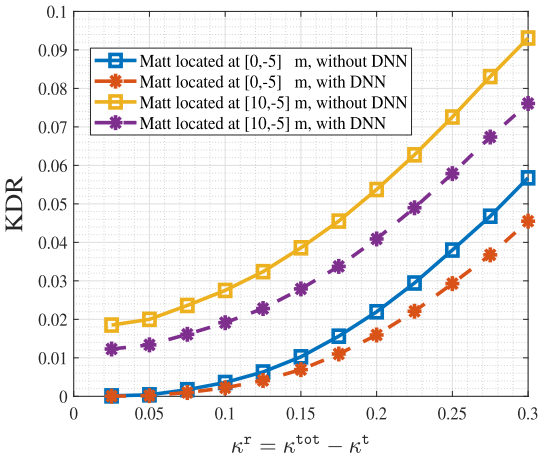

FIGURE 9. Key difference rate vs. the level of receiver impairment for $\rho = 0.95$, $P_{\mathcal{M}} = 10$ dBm, and $P_{\mathcal{A},\mathcal{B},\mathcal{R}} = -10$ dBm.

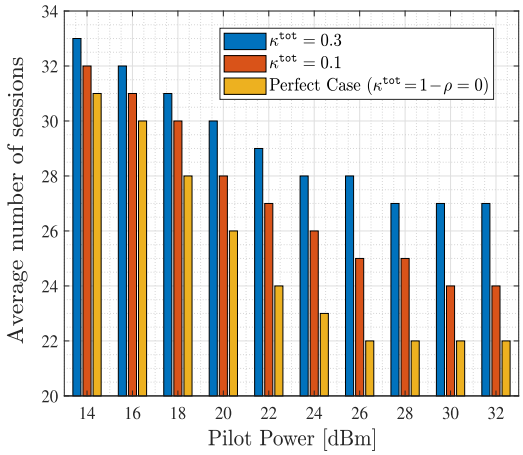

FIGURE 10. Number of required sessions for 256-bit key agreement vs. pilot power for $\rho = 0.95$, $P_{\mathcal{M}} = 10$ dBm, and $\kappa^{\mathrm{t,r}}_{\mathcal{A},\mathcal{B},\mathcal{R}} = \kappa^{\mathrm{tot}}/2$.

In order to generate raw key sequences at Alice and Bob, denoted by $\mathcal{K}_{\mathcal{A}}$ and $\mathcal{K}_{\mathcal{B}}$, respectively, the following one-bit quantization block is employed at Alice and Bob

$$\mathcal{K}_i = \begin{cases} 0 & |\Lambda_i| \leq \mu_{\mathcal{B}} - \Delta\sigma_{\mathcal{B}} \\ 1 & |\Lambda_i| > \mu_{\mathcal{B}} + \Delta\sigma_{\mathcal{B}} \\ \text{none} & \text{otherwise,} \end{cases} \tag{18}$$

where $\Lambda_{\mathcal{A}} = \mathcal{F}_{\mathbf{W},\mathbf{B}}(y_{\mathcal{A}})$ is the predicted data at Alice, and $\Lambda_{\mathcal{B}} = y_{\mathcal{B}}$ is the observation data of Bob at each time-stamp. In addition, $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ denote the mean and standard deviation of Bob's observations, which are assumed to be publicly known among legitimate parties. Moreover, $\Delta = 0.3$ is the quantization guard band [21]. Fig. 9 illustrates the key difference rate (KDR) versus the level of HIs, $\kappa^{\mathrm{r}}_{\mathcal{A}} = \kappa^{\mathrm{r}}_{\mathcal{B}} = \kappa^{\mathrm{r}}$, for two cases of Matt being located at $[0, -5]$m and $[10, -5]$m, respectively. We mention that according to Remark 3, it is not tractable to derive a closed-form expression for the KDR. For this figure, the HI of intermediate relay is set to $[\kappa^{\mathrm{t}}_{\mathcal{R}}, \kappa^{\mathrm{r}}_{\mathcal{R}}] = [0.1, 0.1]$. We have considered that $\kappa^{\mathrm{t}}_{\mathcal{A}(\mathcal{B})} = \kappa^{\mathrm{tot}} - \kappa^{\mathrm{r}}_{\mathcal{A}(\mathcal{B})}$, and $\kappa^{\mathrm{tot}} = 0.3$ [25]. The figure indicates that the level of HIs at receiver ends plays a more important role than the transmit HIs. This can be inferred from the overall proposed protocol in Section III, where the Rx hardware of Alice and Bob contribute in the first and the third step of packet exchange, while their Tx hardware is active in the second step only. Inspecting the residual terms in (11) can also verify this fact. From Fig. 5, one can also infer that when Matt is near one of the legitimate parties, higher levels of mismatch are imposed, leading to higher KDRs. This is because Matt can cause greater discrepancies due to the unbalanced levels of HIs in the adversarial links $\mathbf{h}_{\mathcal{MA}}$ and $\mathbf{h}_{\mathcal{MB}}$. Moreover, Fig. 9 remarks that our proposed DNN is *data efficient* in terms of the values of HIs, i.e., our DNN is able to provide lower KDRs for a wide range of HIs.

Fig. 10 depicts the average number of randomness distillation sessions, i.e., the sessions of packet exchange, required to be performed by $\mathcal{A}$ and $\mathcal{B}$ to agree on a secret key of length $|\mathcal{K}| = 256$ bits [11]. Notably, a key of 256 secure bits can be utilized for encrypting up to gigabytes of data [11]. In this test, the average number of required sessions is calculated based on the general formula of [21], i.e., $\frac{|\mathcal{K}|}{N_{sc}(1-\mathsf{KDR})}$. Moreover, we consider the WSKG scheme over $N_{sc} = 12$ parallel blocks to show the generalization capability of our proposed scheme [18]. The results of Fig. 10 imply that increasing the transmission power of pilot packets can decrease the required number of sessions. This is because increasing pilot power can lower the KDR. For instance, in the ideal case of perfect hardware, by having a transmit power of more than 25 dBm the number of sessions tends to its minimum value of $\frac{|\mathcal{K}|}{N_{sc}} = 22$ which corresponds to the case of one bit quantization with zero KDR. In addition, the figure shows the negative impact of having HIs that can increase the required number of sessions. For instance, having HIs at a level of $\kappa^{\mathrm{tot}} = 0.1$, can impose to the network about 9% increase in the number of required sessions. Thus, it is important to carefully take the hardware and channel imperfections into account to reflect the realistic behavior of wireless systems.

VII. CONCLUSION

In this article, we studied a DL-based approach for relay-aided WSKG scheme in wireless networks under MitM adversarial attacks. We took into account the practical assumptions of HIs and imperfect channel reciprocity to gain realistic understandings of a practical system. To alleviate the MitM from spoofing the randomness distillation, RPs were deployed at PHY layer. We also implemented a DNN, comprised of GRUs, to further improve the WSKG process. The impacts of HIs and MitM adversarial attacks on system's performance were examined, while numerous experiments were conducted to highlight the performance gain of our DL-based approach compared with the state-of-the-arts. Proposing a mathematical framework to analytically study the trade-off between the computation overhead of learning block and the information leakage of the reconciliation phase will be considered in our

future works. Moreover, we will incorporate other types of learning-based attacks, such as adversarial machine learning (AML), into the WSKG process in our future works.

Another important direction that is left for our future work is to study the applications of WSKG scheme at the intersection of 6G networks and emerging technologies such as metaverse and digital twins [46], [47], [48].

REFERENCES

[1] M. Letafati, H. Behroozi, B. H. Khalaj, and E. A. Jorswieck, "Deep learning for hardware-impaired wireless secret key generation with man-in-the-middle attacks," in *Proc. IEEE Glob. Commun. Conf.*, 2021, pp. 1–6.

[2] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[3] A. Chorti et al., "Context-aware security for 6G wireless the role of physical layer security," *IEEE Commun. Standards Mag.*, vol. 6, no. 1, pp. 102–108, Mar. 2022.

[4] J. Zhang, G. Li, A. Marshall, A. Hu, and L. Hanzo, "A new frontier for IoT security emerging from three decades of key generation relying on wireless channels," *IEEE Access*, vol. 8, pp. 138406–138446, 2020.

[5] J. Zhang, S. Rajendran, Z. Sun, R. Woods, and L. Hanzo, "Physical layer security for the Internet of Things: Authentication and key generation," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 92–98, Oct. 2019.

[6] G. Li, C. Sun, J. Zhang, E. Jorswieck, B. Xiao, and A. Hu, "Physical layer key generation in 5G and beyond wireless communications: Challenges and opportunities," *Entropy*, vol. 21, no. 5, pp. 1–16, May 2019.

[7] J. Zhang, A. Marshall, and L. Hanzo, "Channel-envelope differencing eliminates secret key correlation: LoRa-based key generation in low power wide area networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12462–12466, Dec. 2018.

[8] G. Wunder, A. Müller, C. Paar, HD. Schotten, T. Wollinger, and E. A. Jorswieck, "Security made easy for IoT," *Industrie 4.0 Manage.*, vol. 34, pp. 6–9, 2018.

[9] C. Zenger, H. Vogt, J. Zimmer, A. Sezgin, and C. Paar, "The passive eavesdropper affects my channel: Secret-key rates under real-world conditions," in *Proc. IEEE GLOBECOM Workshops*, 2016, pp. 1–6.

[10] M. Letafati, A. Kuhestani, K. -K. Wong, and M. J. Piran, "A lightweight secure and resilient transmission scheme for the internet-of-things in the presence of a hostile jammer," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4373–4388, Mar. 2021.

[11] M. Mitev, A. Chorti, M. Reed, and L. Musavian, "Authenticated secret key generation in delay-constrained wireless systems," *EURASIP J. Wireless Commun. Netw.*, vol. 1, pp. 1–29, Jun. 2020.

[12] S. A. A. Kalkhoran, M. Letafati, E. Erdemir, B. H. Khalaj, H. Behroozi, and D. Gündüz, "Secure deep-JSCC against multiple eavesdroppers," 2023, *arXiv:2308.02892*.

[13] M. Letafati, H. Behroozi, B. H. Khalaj, and E. A. Jorswieck, "On learning-assisted content-based secure image transmission for delay-aware systems with randomly-distributed eavesdroppers," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 1125–1139, Feb. 2022.

[14] H. Boche, R. F. Schaefer, and H. V. Poor, "Algorithmic detection of adversarial attacks on message transmission and ACK/NACK feedback," in *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–6.

[15] M. Letafati, H. Behroozi, B. H. Khalaj, and E. A. Jorswieck, "Hardware-impaired PHY secret key generation with man-in-the-middle adversaries," *IEEE Wireless Comm. Lett.*, vol. 11, no. 4, pp. 856–860, Apr. 2022.

[16] M. Letafati, A. Kuhestani, D. W. K. Ng, and H. Behroozi, "A new frequency hopping-aided secure communication in the presence of an adversary jammer and an untrusted relay," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2020, pp. 1–7.

[17] M. Letafati, A. Kuhestani, H. Behroozi, and D. W. K. Ng, "Jamming-resilient frequency hopping-aided secure communication for internet-of-things in the presence of an untrusted relay," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6771–6785, Oct. 2020.

[18] N. Aldaghri and H. Mahdavifar, "Physical layer secret key generation in static environments," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2692–2705, 2020.

[19] M. Mitev, A. Chorti, E. V. Belmega, and M. Reed, "Man-in-the-middle and denial of service attacks in wireless secret key generation," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.

[20] G. Li, Y. Xu, W. Xu, E. Jorswieck, and A. Hu, "Robust key generation with hardware mismatch for secure MIMO communications," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 5264–5278, 2021.

[21] X. Guan, N. Ding, Y. Cai, and W. Yang, "Wireless key generation from imperfect channel state information: Performance analysis and improvements," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2019, pp. 1–6.

[22] L. Jiao, G. Sun, J. Le, and K. Zeng, "Machine learning-assisted wireless PHY key generation with reconfigurable intelligent surfaces," in *Proc. ACM Workshop Wireless Secur. Mach. Learn.*, 2021, pp. 1–6.

[23] P. Walther and T. Strufe, "Blind twins: Siamese networks for non-interactive information reconciliation," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2020, pp. 1–7.

[24] M. Letafati, H. Behroozi, B. H. Khalaj, and E. A. Jorswieck, "Wireless-powered cooperative key generation for e-health: A reservoir learning approach," in *Proc. IEEE 95th Veh. Technol. Conf.*, 2022, pp. 1–7.

[25] E. Björnson, M. Matthaiou, and M. Debbah, "A new look at dual-hop relaying: Performance limits with hardware impairments," *IEEE Trans. Commun.*, vol. 61, no. 11, pp. 4512–4525, Nov. 2013.

[26] K. Sankhe et al., "No radio left behind: Radio fingerprinting through deep learning of physical-layer hardware impairments," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 165–178, Mar. 2020.

[27] T. Schenk, *RF Imperfections in High-Rate Wireless Systems: Impact and Digital Compensation*. Berlin, Germany: Springer, 2008.

[28] enisa.europa.eu, "Man-in-the-middle," European Union Agency for Cybersecurity (ENISA), 2021. Accessed: Sep. 24, 2023. [Online]. Available: https://www.enisa.europa.eu/topics/csirts-in-europe/glossary/man-in-the-middle

[29] C. D. T. Thai, J. Lee, and T. Q. S. Quek, "Physical-layer secret key generation with colluding untrusted relays," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1517–1530, Feb. 2016.

[30] H. -M. Wang, K. Huang, and T. A. Tsiftsis, "Multiple antennas secure transmission under pilot spoofing and jamming attack," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 4, pp. 860–876, Apr. 2018.

[31] N. O'Donoughue and J. M. F. Moura, "On the product of independent complex Gaussians," *IEEE Trans. Signal Process.*, vol. 60, no. 3, pp. 1050–1063, Mar. 2012.

[32] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 6645–6649.

[33] A. Graves, "Generating sequences with recurrent neural networks," 2013, *arXiv:1308.0850*.

[34] Q. Wang, H. Li, D. Zhao, Z. Chen, S. Ye, and J. Cai, "Deep neural networks for CSI-based authentication," *IEEE Access*, vol. 7, pp. 123026–123034, 2019.

[35] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[36] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light gated recurrent units for speech recognition," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 2, pp. 92–102, Apr. 2018.

[37] C. Xu, J. Shen, X. Du, and F. Zhang, "An intrusion detection system using a deep neural network with gated recurrent units," *IEEE Access*, vol. 6, pp. 48697–48707, 2018.

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[39] D. P. Kingma and L. J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–13.

[40] M. Letafati, H. Behroozi, B. H. Khalaj, and E. A. Jorswieck, "Content-based medical image transmission against randomly-distributed passive eavesdroppers," in *Proc. IEEE Int. Conf. Commun. Workshop*, 2021, pp. 1–7.

[41] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7957–7969, Aug. 2019.
[42] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2006.
[43] C. J. Cellucci, A. M. Albano, and P. E. Rapp, "Statistical validation of mutual information calculations: Comparison of alternative numerical algorithms," *Amer. Phys. Soc.*, vol. 71, no. 6, Jun. 2005, Art. no. 066208.
[44] Q. Xie, Z. Wang, and Z. Yang, "Polar decomposition of mutual information over complex-valued channels," *IEEE Trans. Inf. Theory*, vol. 60, no. 6, pp. 3163–3171, Jun. 2014.
[45] P. J. Freire, S. Srivallapanondh, A. Napoli, J. E. Prilepsky, and S. K. Turitsyn, "Computational complexity evaluation of neural network applications in signal processing," Jun. 2022, *arXiv:2206.12191v1*.
[46] M. Letafati and S. Otoum, "On the privacy and security for e-health services in the metaverse: An overview," *Ad Hoc Networks*, vol. 150, Nov. 2023, doi: 10.1016/j.adhoc.2023.103262.
[47] M. Letafati and S. Otoum, "Digital healthcare in the metaverse: Insights into privacy and security," Aug. 2023, *arXiv:2308.04438v2*.
[48] M. Letafati and S. Otoum, "Global differential privacy for distributed metaverse healthcare systems," Aug. 2023, *arXiv:2308.04439v2*.

MEHDI LETAFATI (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2019 and 2021, respectively. He is currently a Doctoral Researcher with the Center for Wireless Communications, University of Oulu, Oulu, Finland, with the 6G Flagship project. Before that, he was with the College of Technological Innovation, Zayed University, Abu Dhabi, UAE, working on metaverse healthcare and privacy as a Research Assistant. In August 2020, he was a program attendee with Cornell, Maryland, Max-Planck Pre-Doctoral Research School, Saarbrüecken, Germany. His research interests include the intersection of machine learning and wireless communications. He is also interested in information theory, data science, and enabling technologies for digital healthcare and Metaverse. He was honored to be ranked 4th among all participants in the Nationwide University Entrance Exam in 2015. He was the recipient of the Exceptional Talent for outstanding performance during his undergraduate and Master studies, and National Elite Foundation's scholarships for teaching assistance. Mehdi is a peer reviewer for top-tier IEEE journals, including IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.

HAMID BEHROOZI (Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2000, the M.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, in 2003, and the Ph.D. degree in electrical engineering from Concordia University, Montreal, QC, Canada, in 2007. From 2007 to 2010, he was a Postdoctoral Fellow with the Department of Mathematics and Statistics, Queen's University, Kingston, ON, Canada. He is currently an Associate Professor with the Department of Electrical Engineering, Sharif University of Technology. His research interests include information theory, joint source-channel coding, artificial intelligence in signal processing and data science, and cooperative communications. He was the recipient of several academic awards, including the Ontario Postdoctoral Fellowship awarded by the Ontario Ministry of Research and Innovation (MRI), Quebec Doctoral Research Scholarship awarded by the Government of Quebec (FQRNT), Hydro Quebec Graduate Award, and Concordia University Graduate Fellowship.

BABAK HOSSEIN KHALAJ (Senior Member, IEEE) received the B.Sc. degree in electrical Engineering from the Sharif University of Technology, Tehran, Iran, in 1989, and the M.Sc. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1993 and 1996, respectively. He has been with the pioneering team with Stanford University, where he was involved in adoption of multi-antenna arrays in mobile networks. Since 1999, he has been a Senior Consultant in the area of data communications, and a Visiting Professor with CEIT, San Sebastian, Spain, from 2006 to 2007. He has coauthored many papers in signal processing and digital communications and holds four U.S. patents. He was the recipient of the Alexander von Humboldt Fellowship from 2007 to 2008 and Nokia Visiting Professor Scholarship in 2018.

EDUARD A. JORSWIECK (Fellow, IEEE) received the Ph.D. degree in electrical engineering and computer science from TU Berlin, Berlin, Germany, in 2004. He is currently the Managing Director of the Institute of Communications Technology and the Head of the Chair of communications Systems and a Full Professor with the Technische Universität Braunschweig, Brunswick, Germany. From 2008 to 2019, he held the Chair of communication theory with TU Dresden. From 2006 to 2008, he was with the Signal Processing Group with KTH Stockholm as a Postdoctoral Fellow and Assistant Professor. He has coauthored more than 170 journal articles, 15 book chapters, one book, three monographs, and some 300 conference papers. His main research focuses on the broad area of communications. He was the recipient of the IEEE Signal Processing Society Best Paper Award. He and his colleagues were also recipients of the Best Paper and Best Student Paper Awards at the IEEE CAMSAP 2011, IEEE WCSP 2012, IEEE SPAWC 2012, IEEE ICUFN 2018, PETS 2019, and ISWCS 2019. Since 2017, he has been the Editor-in-Chief of the Springer *EURASIP Journal on Wireless Communications and Networking*. Since 2022, he has been on the Editorial Board of IEEE TRANSACTIONS ON COMMUNICATIONS. He was on the Editorial Boards of IEEE SIGNAL PROCESSING LETTERS, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.