

Deep-Learning-Assisted IoT-Based RIS for Cooperative Communications

Bulent Sagir¹, Erdogan Aydin², and Hacı İlhan³, *Senior Member, IEEE*

Abstract—Reconfigurable intelligent surfaces (RISs) are software-controlled passive devices that can be used as relay (R) systems to reflect incoming signals from a source (S) to a destination (D) in a cooperative manner with optimum signal strength to improve the performance of wireless communication networks. The configurability and flexibility of an RIS deployed in an Internet of Things (IoT)-based network can enable network designers to devise stand-alone or cooperative configurations that have considerable advantages over conventional networks. In this article, two new deep neural network (DNN)-assisted cooperative RIS (CRIS) models, namely, DNN_R - CRIS and $DNN_{R,D}$ - CRIS, are proposed for cooperative communications. In DNN_R - CRIS model, the potential of RIS deployment as an IoT-based relay element in a next-generation cooperative network is investigated using deep-learning (DL) techniques for RIS phase optimization. In addition, to reduce the maximum-likelihood (ML) complexity at D, a new DNN-based symbol detection method is presented with the $DNN_{R,D}$ - CRIS model combined with DNN-assisted phase optimization. For a different number of relays and receiver configurations, the bit error rate (BER) performance results of the proposed DNN_R - CRIS and $DNN_{R,D}$ - CRIS models and traditional CRIS scheme (without a DNN) are presented for a multirelay cooperative communication scenario with path loss effects. It is revealed that the proposed DNN-based models show promising results in terms of BER, even in high-noise environments with low system complexity.

Index Terms—Bit error rate (BER), cooperative communication, deep learning (DL), deep neural network (DNN), Internet of Things (IoT), machine learning, reconfigurable intelligent surface (RIS), relaying.

I. INTRODUCTION

FORTHCOMING 5G and more advanced wireless network architectures promise a several fold increase in transmission capacity and much lower latency levels compared to current wireless networks by introducing new radio spectra, including millimeter waves, and a bundle of emerging technologies, such as massive multiple-input-multiple-output (MIMO), small cells, edge computing, device centric architectures, and new channel coding techniques [1], [2].

Manuscript received 15 June 2022; revised 17 November 2022 and 30 December 2022; accepted 13 January 2023. Date of publication 25 January 2023; date of current version 7 June 2023. (Corresponding author: Erdogan Aydin.)

Bulent Sagir is with the Wholesale Business Unit, Türk Telekomünikasyon A.Ş., 34660 Istanbul, Turkey (e-mail: bulent.sagir@turktelekom.com.tr).

Erdogan Aydin is with the Department of Electrical and Electronics Engineering, Istanbul Medeniyet University, 34857 Istanbul, Turkey (e-mail: erdogan.aydin@medeniyet.edu.tr).

Hacı İlhan is with the Department of Electronics and Communications Engineering, Yıldız Technical University, 34220 Istanbul, Turkey (e-mail: ilhanh@yildiz.edu.tr).

Digital Object Identifier 10.1109/JIOT.2023.3239818

The expected improvements in the physical layer apart from the new spectrum usage depend largely on the control of the propagation environment with techniques, such as employing antenna diversity in transmitters and receivers, cooperative relaying schemes, and intelligent surfaces. Among these, reconfigurable intelligent surfaces (RISs), which may also be referred to as software-defined surfaces (SDSs) or large intelligent surfaces (LISs) in the literature, is a brand new concept that enables every surface in a propagation environment to be used effectively to control the environment for efficient transmission and even to exploit the scattering waves [3], [4]. An RIS is a thin surface of multiple reflecting elements made of low-cost passive electronic devices to alter the incident wave phase angles. This concept was first considered in [5] in 2012, where intelligent walls with active frequency selective surfaces were proposed for controlling the propagation environment in indoor scenarios. In [6], LIS was proposed to form a contiguous surface of an electromagnetically active material as an alternative to massive MIMO.

Although intelligent surfaces were initially mainly intended for indoor environments, in [5], [6], [7], and [8], these surfaces were used in outdoor scenarios in the form of passive intelligent mirrors (PIMs) for multiuser multiple-input-single-output (MISO) downlink communication [9]; PIMs could be deployed around building exteriors to replace base stations (BSs) serving as mobile terminals. The outdoor usage scenarios can be diversified with the progress in material technology.

As an RIS comprises low-cost passive reflectors, one of its significant benefits is that it can be used in a relatively high-cost next-generation network. Its ability to be applied on various surfaces allows it to be used in diverse cases. In [10], an RF transmitter was deployed in tandem with an RIS for modulating an RF signal, and the RIS was effectively transformed into an RF modulator. Furthermore, various applications of RISs, such as using RISs to create a virtual line-of-sight (LOS) link between BSs and users where it is not feasible to establish a direct reliable link, improving the physical layer security in a wireless network, or realizing simultaneous wireless information and power transfer (SWIPT) in an Internet of Things (IoT) network [8], [11], have been investigated. Additionally, RIS-based space shift keying (SSK), spatial modulation (SM), and media-based modulation (MBM) schemes were proposed in [12] and [13].

Another noteworthy approach is the concept of software-controlled hypersurfaces, which interact with electromagnetic waves in a controlled manner to re-engineer the impinging waves to obtain the desired response [14], [15]. In the proposed

model, each hypersurface is made of intelligent surfaces consisting of ultrathin meta atoms that receive commands from a centralized controller and adjust their EM behavior through an IoT gateway. The software-configurable nature of hypersurfaces makes it possible to deploy each unit as an IoT device in a broader network structure.

For current and future wireless network architectures, using relay-based cooperative communication techniques to create a link or route diversity is an emerging research area for achieving the desired coverage and throughput capacities [16], [17], [18]. In a typical cooperative system, single or multiple relays are deployed between source (S) and destination (D) nodes to create alternative routes for signal transmission [19]. Consequently, for these types of networks, the possibility of using an RIS as a relay (R) should be considered, as both devices have very similar functions, e.g., to create route diversity, and turn non-LOS links into LOS links. This is one of the active research areas where the performance and cost of both methods have been investigated and compared. Although relaying is a well-established method to achieve this, the requirements can become too steep to be applied in a field where each relay requires a dedicated power supply and front-end circuitry, which consequently increases the total power requirements, costs, and implementation complexity. One of the main issues discussed extensively in [19] was addressed in [20] for wireless networks operating in high-frequency bands. The power consumption and energy efficiency of decode-and-forward (DF) and amplify-and-forward (AF) relays against RISs were compared, and it was shown that sufficiently large RISs could outperform relay supported systems in terms of data rates, with reduced implementation complexities [21].

Using deep learning (DL) or machine learning methods to optimize RIS utilization in wireless networks is a brand new approach and has been addressed in several studies, some of which focus on subjects, such as symbol estimation [22], cooperative communication [23], indoor signal focusing [24], or securing wireless communication [25], [26]. The results achieved with DL techniques often show better or at least comparable performance against conventional methods while significantly reducing the computational complexity.

In RIS-assisted communication systems, channel state information (CSI) estimation can be a challenging process, and a great deal of the given studies so far considers the CSI is perfectly known at the RIS or receiver. However, CSI estimation is possible with techniques already covered in various works [27], [28], beside some other studies dealing with totally blind or imperfect CSI conditions [10], [22], [29]. Deploying DL-based estimation techniques is shown to provide promising results, which can be a groundwork for future research.

A. Contributions

In this work, we investigate the possibility of a novel usage scenario: using an RIS as the sole relaying element in a cooperative network, where multiple DL-optimized RISs are utilized in a cooperative configuration between S and D . Our work focuses exclusively on the outdoor scenario, where building facades are covered with RISs in a dense urban

environment with slow fading channels. In our proposed DL-assisted cooperative RIS (CRIS) models ($\text{DNN}_R - \text{CRIS}$ and $\text{DNN}_{R,D} - \text{CRIS}$), RISs are optimized through a deep neural network (DNN), which is trained with the CSI obtained for incident and reflected signals to estimate the phase adjustments required for optimal signal transmission of RISs. Furthermore, in the $\text{DNN}_{R,D} - \text{CRIS}$ model, instead of using a conventional maximum-likelihood (ML) detector at D , we deploy another DNN to estimate the received symbols. Using multiple DNNs on different parts of the network, it finally becomes possible to obtain an approximation of an end-to-end DNN-reinforced communication network and to compare its performance against conventional network architectures.

In the proposed DNN optimized model, all DNNs deployed on the RISs are considered to be embedded in the RIS controller hardware and interconnected through an IoT network to communicate and coordinate with other network components. Here, IoT integration is supposed to provide a control platform for all RIS-based relays to supply the system with network and problem management capabilities along with other functional capabilities, such as transmission through relay selection.

The performances of the $\text{DNN}_R - \text{CRIS}/\text{DNN}_{R,D} - \text{CRIS}$ models are analyzed in terms of the bit error rate (BER) for an M -array quadrature amplitude modulation (QAM) scheme using various relay configurations with path loss effects. Furthermore, a complexity analysis is also included, comparing DNN-assisted and conventional methods. Simulating and measuring the effects of multiple RISs on the received signal in a dense urban environment are the main goals of our work; consequently, the potential of DNN-assisted RIS usage in a next-generation wireless network is exhibited.

For all the scenarios and configurations covered in this work, it is assumed that the CSI is perfectly known at the IoT-based relays interconnected through the IoT network, gathering full CSI data extracted from the pilot signals.

As a result of the analyses, we can summarize the contributions of this article as follows.

- 1) In the proposed $\text{DNN}_R - \text{CRIS}$ and $\text{DNN}_{R,D} - \text{CRIS}$ models, RISs deployed as IoT-based relay elements in a cooperative network are presented. For the RIS-based relaying model, it is possible to select the best performing R among multiple relay configurations between S and D with a choice of conventional relay selection (RS) methods.
- 2) RIS-based relays are conceived as IoT devices communicating in a broader IoT network where DNN input signals and other vital signaling are transmitted through IoT gateways. This kind of IoT integration for RIS-based relays in a cooperative communications scenario is also limited in the literature.
- 3) In both models, RIS configurations are optimized with DNNs. A DNN is deployed for controlling each RIS in the system and to optimize the phase adjustments in real time. In the $\text{DNN}_{R,D} - \text{CRIS}$ model, an additional DNN is deployed at the destination for symbol detection instead of an ML detector. The performance results of the ML detector and DNN at the destination are compared in terms of BER. Additionally, through

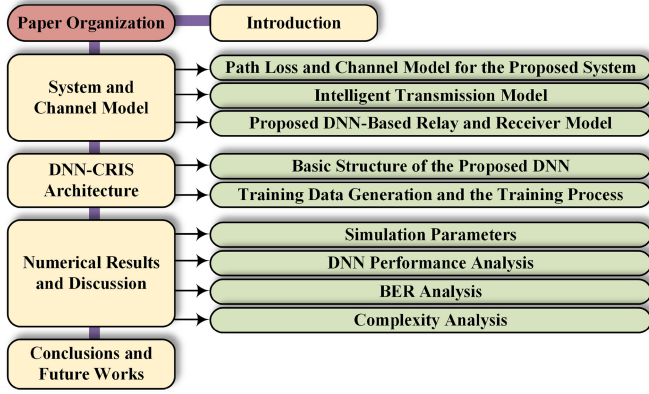


Fig. 1. Organization of this article.

this model, DNN-assisted phase and symbol detection are collectively deployed and analyzed in a cooperative network.

- 4) In an RIS-based multiple relay case, the effect of the relay locations on receiver performance with path loss effects is also investigated. The distance from the relays to S or D in conjunction with fading channels has a critical impact on BER performance, and this is demonstrated extensively in our work.
- 5) An extensive evaluation of the results is also performed with a complexity analysis, comparing proposed DNN-assisted and conventional methods.

B. Organization and Notations

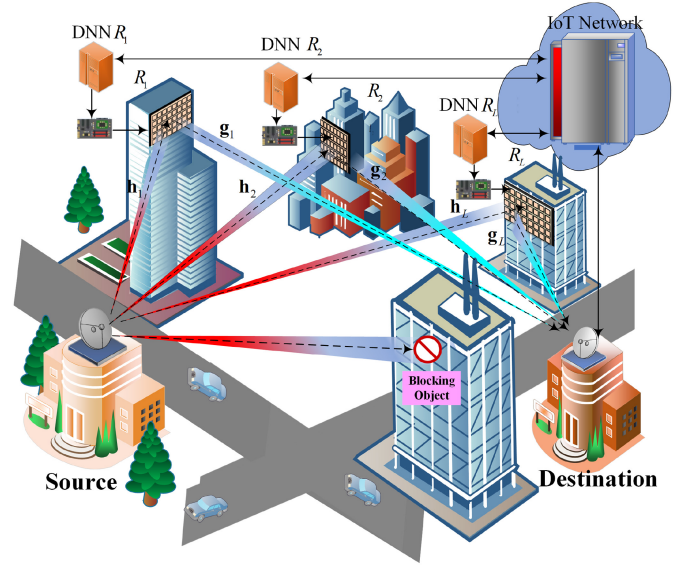
The organization of this article is given in Fig. 1. The system and channel models are presented in Section II, with the theoretical background of transmission and the proposed DNN-based relay and receiver models. In Section III, the architecture of the proposed model with details of the basic structure of the proposed DNNs, training data generation and training processes are thoroughly covered for both DNNs. Section IV provides simulation setup and performance results for a series of RIS configurations, including a detailed performance comparison of the proposed model against various parameters. Finally, Section V concludes this article.

Notations: The following notations are used throughout this article: 1) bold lower/upper case symbols represent vectors/matrices; 2) $(\cdot)^T$ and $\|\cdot\|_F$ denote transpose and Frobenius norm operators, respectively; 3) $\Re(\cdot)$ and $\Im(\cdot)$ are the real and imaginary parts of a complex-valued quantity; 4) $(\cdot)^*$ represents the conjugate operator; 5) $[\cdot]_{a \times b}$ indicates the dimensions of a matrix with a rows and b columns; and 6) $O(\cdot)$ represents the time or computational complexity of an algorithm, in terms of number of operations.

II. TRANSMISSION THROUGH

DNN_R – CRIS/DNN_{R,D} – CRIS: SYSTEM MODEL

In this section, we provide an overview of the generic model of the proposed DNN_R – CRIS and DNN_{R,D} – CRIS schemes. The system model under consideration will be

Fig. 2. System model of the DNN_R – CRIS/DNN_{R,D} – CRIS.

named as DNN_R – CRIS if a DNN is used in R only, and DNN_{R,D} – CRIS if DNN is used in both R and D .

System model for the proposed scheme is shown in Fig. 2, where S and D contain a single transmit and receive antennas. In the considered scenario, assuming no LOS link between S and D , cooperative transmission occurs only through multiple RIS elements. We assume that each RIS has the form of a reflect-array comprising N simple and reconfigurable reflector elements and acts as an R_ℓ in a cooperative network routing the transmission from S to D . In the proposed system model, all RISs are assumed to be individually controlled and configured by an IoT-based controller, which hosts both management and DNN software for the RIS. Software-based DNN constantly computes the phase shifts for each reflector and adjusts the configurations for the reflector elements accordingly. Through the IoT network, each controller can be conceived as a gateway for forwarding input signals to the DNN and managing the RIS-based relay for additional functionality, such as activation/deactivation, fault management, troubleshooting, software updates, etc. In the proposed model, each RIS is considered as an IoT device where the existing IoT platforms and communication protocols can be deployed, details of which will be beyond the context of this work. The software-based DNN architecture will be covered in detail in subsequent sections.

At the destination end, two scenarios are considered. First, an ML-detector is deployed to estimate the received symbols (DNN_R – CRIS). Second, another DNN is used to estimate the received symbols implemented in the receiver hardware, effectively replacing an ML-detector (DNN_{R,D} – CRIS). We will evaluate and compare the performances of both detection schemes in Section IV.

A. Path Loss and Channel Model for the Proposed System

For the transmission model, we assume that S , D , and each R_ℓ are positioned in a triangle formation in a 2-D space as shown in Fig. 3, where d_{SD} , d_{SR_ℓ} , and $d_{R_\ell D}$ represent the

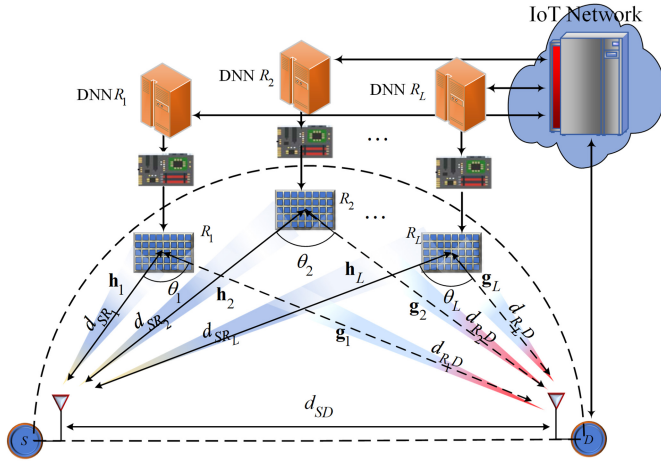


Fig. 3. Channel model for DNN-CRIS system.

distances of source to destination ($S \rightarrow D$), source to ℓ th relay ($S \rightarrow R_\ell$), and ℓ th relay to destination ($R_\ell \rightarrow D$), respectively, where $\ell = 1, 2, \dots, L$ and here L is the number of relays. In this configuration, S and D stay at the either end of the triangle base and R_ℓ is positioned on the top corner such that all three corners are kept within the half circle as shown in Fig. 3, assuming $\pi/2 < \theta_\ell < \pi$, where θ_ℓ is the angle between d_{SR_ℓ} and $d_{R_\ell D}$. In this formation, R_ℓ lies on the circle arc when $\theta_\ell = \pi/2$, and lies on the line $S \rightarrow D$ when $\theta = \pi$. In all cases, d_{SD} will be equal to the diameter.

In the proposed channel model, all links are assumed to be exposed to both long-term free space path loss and short-term Rayleigh fading. Here, the path loss is proportional to d^{-c} , where d is the propagation distance and c is the path loss exponent [30]. As the path loss for the channel $S \rightarrow D$ is assumed to be unity, the relative gains of $S \rightarrow R_\ell$ and $R_\ell \rightarrow D$ can be defined as $G_{SR_\ell} = (d_{SD}/d_{SR_\ell})^c$ and $G_{R_\ell D} = (d_{SD}/d_{R_\ell D})^c$, respectively. Here, G_{SR_ℓ} and $G_{R_\ell D}$ are related by law of cosines [31], which is

$$G_{SR_\ell}^{-2/c} + G_{R_\ell D}^{-2/c} - 2G_{SR_\ell}^{-1/c} G_{R_\ell D}^{-1/c} \cos \theta_\ell = 1. \quad (1)$$

Through (1), assuming d_{SR_ℓ} and θ_ℓ are given and $d_{SD} = 1$, $d_{R_\ell D}$ can be computed in order to apply path loss to complex channel coefficients between $S \rightarrow R_\ell$ and $R_\ell \rightarrow D$, which can be expressed as $\mathbf{h}_\ell = [h_{\ell,1}, h_{\ell,2}, \dots, h_{\ell,N}] \in \mathbb{C}^{1 \times N}$ and $\mathbf{g}_\ell = [g_{\ell,1}, g_{\ell,2}, \dots, g_{\ell,N}] \in \mathbb{C}^{1 \times N}$, where $\ell = 1, 2, \dots, L$ and $\mathbb{C}$ denotes the set of complex numbers. Channel vector elements can be expressed in terms of channel amplitudes and phases $h_{i,j} = \alpha_{i,j} e^{-j\phi_{i,j}}$ and $g_{i,j} = \beta_{i,j} e^{-j\theta_{i,j}}$, respectively, where $i \in \{1, 2, \dots, L\}$ and $j \in \{1, 2, \dots, N\}$. In the considered system structure, all channel coefficients are assumed to be Rayleigh fading distribution. Under this assumption, we have $h_{i,j}, g_{i,j} \sim CN(0, 1)$, where $h_{i,j}, g_{i,j} \sim CN(0, \sigma^2)$ stands for the complex Gaussian distribution with zero mean and σ^2 variance.

With this model, it is possible to observe the effect of various relay positionings and evaluate their performances.

B. Intelligent Transmission Model

The noisy received baseband signals reflected through the ℓ th relay with N passive elements can be expressed at the D as follows:

$$\begin{aligned} y_\ell &= \sqrt{G_{SR_\ell} G_{R_\ell D}} \left(\sum_{n=1}^N h_{\ell,n} e^{j\phi_{\ell,n}} g_{\ell,n} \right) x + w_\ell \\ &= \sqrt{G_{SR_\ell} G_{R_\ell D}} \left(\mathbf{h}_\ell \Phi_\ell \mathbf{g}_\ell^T \right) x + w_\ell. \end{aligned} \quad (2)$$

Here, x is the modulated M -QAM signal, and $w_\ell \sim CN(0, N_0)$ represents the additive white Gaussian noise (AGWN) with zero mean and $N_0/2$ variance per dimension at the receiver [32]. $\Phi_\ell \triangleq \text{diag}(\phi_\ell) \in \mathbb{C}^{N \times N}$ is the diagonal phase matrix for each RIS including the adjusted phase angles by RIS reflecting elements, where the phase vector for ℓ th RIS is defined as $\phi_\ell = [\phi_{\ell,1}, \phi_{\ell,2}, \dots, \phi_{\ell,N}]$. The vectorial representation of the noisy baseband signals reflected through all relays with N passive elements can be expressed at D as follows:

$$\mathbf{y} = [y_1, y_2, \dots, y_L]^T \in \mathbb{C}^{L \times 1}. \quad (3)$$

From (2), we can find the instantaneous signal-to-noise ratio (SNR) at D for the reflected signals through the ℓ th relay with N passive elements as

$$\gamma_\ell = \frac{\left| \sum_{i=1}^N \alpha_{\ell,i} \beta_{\ell,i} e^{j(\phi_{\ell,i} - \phi_{\ell,i} - \theta_{\ell,i})} \right|^2 G_{SR_\ell} G_{R_\ell D} E_s}{N_0} \quad (4)$$

where E_s is the average transmitted energy per symbol. Adjustment of the phase angles by RIS is based on the fact that the maximum signal strength at D can be achieved when $\phi_{\ell,i} = \phi_{\ell,i} + \theta_{\ell,i}$ for $i = 1, \dots, N$, assuming the channel phases are known at the RIS. This can be verified by the identity $|\sum_{i=1}^N v_i e^{j\xi_i}|^2 = \sum_{i=1}^N v_i^2 + 2 \sum_{i=1}^N \sum_{t=i+1}^N v_i v_t \cos(\xi_i - \xi_t)$ where maximum value is achieved when $\xi_i = \xi_t$ for all i . Using this fact for phase adjustment at each reflecting element, the maximum instantaneous SNR is achieved at D for the reflected signals through the ℓ th relay with N passive elements as

$$\gamma_\ell = \frac{\left| \sum_{i=1}^N \alpha_{\ell,i} \beta_{\ell,i} \right|^2 G_{SR_\ell} G_{R_\ell D} E_s}{N_0} = \frac{\chi^2 G_{SR_\ell} G_{R_\ell D} E_s}{N_0} \quad (5)$$

where $\chi = |\sum_{i=1}^N \alpha_{\ell,i} \beta_{\ell,i}|$. The total instantaneous SNR (γ_T) at D for the reflected signals through all relays with N passive elements can be written as $\gamma_T = \sum_{\ell=1}^L \gamma_\ell$.

Finally, the achievable throughput performance of the considered DNN_R - CRIS/DNN_{R,D} - CRIS communication systems is given by $\mathcal{R} := \log_2(1 + \gamma_T)$.

1) *Combining Techniques at the Destination*: Combining techniques used at D are as follows.

a) *Transmitting through the best performing relay*: In this technique, the relay that provides the lowest BER value is selected among L relays for transmission. In this case, the best RS criterion will be

$$\widetilde{\text{BER}}_{\ell^*} = \arg \min_{\ell} \{\text{BER}_\ell\}, \quad \ell = 1, 2, \dots, L. \quad (6)$$

Here, ℓ^* denotes the relay which provides the lowest BER value among $\ell = 1, 2, \dots, L$ relays [32].

b) *Maximum ratio combining*: Using Maximum Ratio Combining (MRC) technique for combining the signals received through multiple relays, the combined overall signal at D can be written as

$$y_{MRC} = \zeta_1 y_1 + \zeta_2 y_2 + \dots + \zeta_L y_L \quad (7)$$

where y_ℓ is defined in (2). Since the product of a complex number with its conjugate cancels the phase and results with the sum of the squared value of the magnitudes, the combining coefficient ζ_ℓ should be conjugate of the complex channel response so that the product will yield absolute squares of the channel responses. Thus, the combining coefficient ζ_ℓ for the ℓ th relay will be expressed as follows [32]:

$$\zeta_\ell = \left(\sum_{n=1}^N h_{\ell,n} e^{j\phi_{\ell,n}} g_{\ell,n} \right)^* = (\mathbf{h}_\ell \mathbf{\Phi}_\ell \mathbf{g}_\ell^T)^*. \quad (8)$$

After MRC-based combining is carried out at the receiver, ML detection applied on the combined signal can be expressed as

$$\hat{x}_{MRC} = \arg \min_{v \in \{1, 2, \dots, M\}} \left\{ \left| y_{MRC} - \left(\zeta_1 \sum_{n=1}^N h_{1,n} e^{\hat{\phi}_{1,n}} g_{1,n} + \zeta_2 \sum_{n=1}^N h_{2,n} e^{\hat{\phi}_{2,n}} g_{2,n} + \dots + \zeta_L \sum_{n=1}^N h_{L,n} e^{\hat{\phi}_{L,n}} g_{L,n} \right) x_v \right|^2 \right\}. \quad (9)$$

Here, M is the modulation order and x_v is the symbol to be transmitted, where $v \in 1, 2, \dots, M$.

c) *ML detector*: For the first scenario in the proposed system model, an ML detector is deployed at D for symbol detection. Using the ML detector, the estimated symbol at the receiver can be defined as [31]

$$\hat{x}_{ML} = \arg \min_{v \in \{1, 2, \dots, M\}} \left\{ \left\| \mathbf{y} - \left[\sum_{n=1}^N h_{1,n} e^{\hat{\phi}_{1,n}} g_{1,n} + \sum_{n=1}^N h_{2,n} e^{\hat{\phi}_{2,n}} g_{2,n} + \dots + \sum_{n=1}^N h_{L,n} e^{\hat{\phi}_{L,n}} g_{L,n} \right]^T x_v \right\|^2 \right\}. \quad (10)$$

C. Proposed DNN-Based Relay and Receiver Model

This section explains how DNN-assisted RIS technique can be implemented in an RIS-assisted cooperative communication scenario.

Artificial neural networks (ANNs) are mathematical structures inspired by the biological neural networks consisting of interconnected neurons to form a network that can be trained to perform desired tasks. A DNN can be realized with an ANN with multiple hidden layers with a much greater function approximation ability than a single hidden layer network [33]. In the proposed model, DNNs have been deployed at two critical locations: 1) at the relays to adjust the phase shifts for each reflector and 2) at D to estimate the received symbols.

As stated earlier, each DNN-based relay or RIS in the system is considered to be equipped with an IoT-based controller unit responsible for controlling the states of reflecting

elements on the RIS to adjust the reflection phase shifts for outgoing signals. In this respect, proposed DNN software will be implemented in the controller hardware. The controller is also responsible for communicating with the outer world, performing as a gateway for network management functions, and last but not least, for receiving CSI data to be fed into the DNN, critical for both training and transmission phases, which will be discussed in the following section.

1) *DNN_R – CRIS Model*: In this model, a DNN is deployed in each R , pretrained with a sample set of channel coefficients and respective phase adjustment information ($\mathbf{h}_\ell, \mathbf{g}_\ell, \mathbf{\Phi}_\ell$) based on the theoretical framework provided in the next section. The objective of the proposed DNN-based system is to compute the optimal phase vector that maximizes reflected signals from the relays. The trained DNN dynamically updates the phase adjustments on each RIS reflecting element according to channel information varying in time. Hence, the whole process is divided into two phases.

1) *Training Phase*: As shown in Fig. 4, the training data set is created during training phase, containing $S \rightarrow R_\ell$ and $R_\ell \rightarrow D$ channel coefficients and the corresponding optimum phase configurations (i.e., $\mathbf{h}_\ell, \mathbf{g}_\ell, \mathbf{\Phi}_\ell$, where $\ell = 1, 2, \dots, L$). Here, the CSI is assumed to be perfectly obtained by the pilot signals through IoT gateways, and preprocessed by the controller hardware to compile the training data set. After the data set is created, the elements of this data set will be used to train the DNNs to find optimum RIS phase vectors for each R , and consequently, weight and bias coefficients ($\mathbf{W}_k, \mathbf{b}_k$) of DNN will be updated.

2) *Transmission Phase*: During actual transmission, DNN receives the new $S \rightarrow R_\ell$ and $R_\ell \rightarrow D$ channel coefficients and computes the adjusted phase vector and forwards it to the RIS. The RIS adopts the computed phase configuration to assist in the transmission until new channel coefficients received.

2) *DNN_{R,D} – CRIS Model*: In the second scenario, another DNN is deployed at D instead of an ML detector to detect the received symbols. Like the preceding DNNs at the relay stage, the process is divided into training and transmission phases, where the network is first trained using the appropriate training data set. Then, the trained network carries out the symbol detection based on the noisy signals received at D during the transmission phase.

1) *Training Phase*: The training data set contains the noisy and path faded signals received at D as feature vectors and the respective modulated symbols as outputs. The objective of the training process is to update the $\mathbf{W}_k$ and $\mathbf{b}_k$ parameters of DNN such that for every received signal, DNN converges to the desired symbol.

2) *Transmission Phase*: The DNN at D with updated parameters $\mathbf{W}_k$ and $\mathbf{b}_k$ estimates the correct symbols from the noisy signals reaching D in real time during the actual transmission.

Here, the detection process at D will be defined as a multiclass classification problem such that the elements of the symbol (constellation) set determined by modulation order are

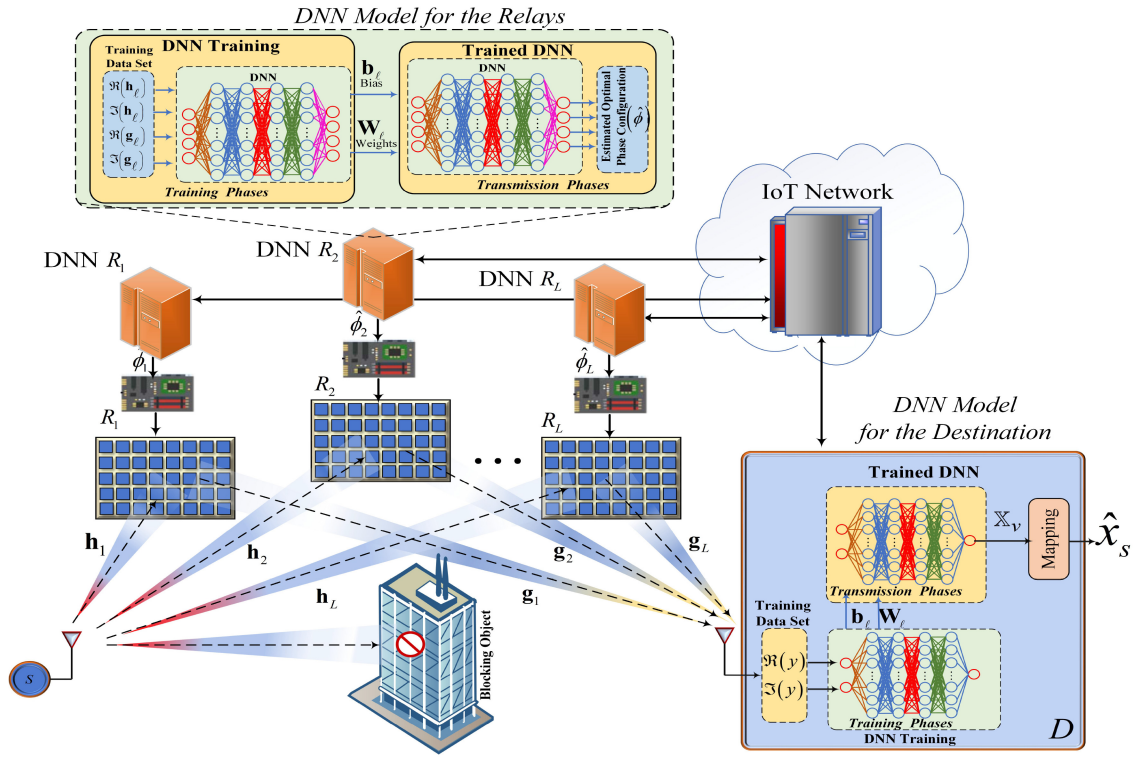


Fig. 4. Training and transmission phases of the proposed DL-assisted CRIS scheme.

mapped to classes to be estimated by the DNN. The details of the process will be covered in the following sections.

III. DNN_R – CRIS/DNN_{R,D} – CRIS ARCHITECTURE

This section presents the architectural details of the proposed DNN schemes to find the optimal phase adjustments for RIS-based relays and to detect symbols at D . The proposed scenario is based on a double DNN architecture as shown in Fig. 4, where the first DNN is deployed at R for estimating the optimal phase vectors using $S \rightarrow R_\ell$ and $R_\ell \rightarrow D$ channel coefficients, and the second DNN is deployed at D to detect the received symbols using path faded and noisy signals at D . The sections below explain our analysis of the proposed system's structure, training procedure, and activation function. As the model is based on a double DNN architecture, both the common and specific structural properties of both DNNs will be pointed out in the next section for convenience.

A. Basic Structure of the Proposed DNNs

Let $\Theta \triangleq \{\theta_1, \theta_2, \dots, \theta_K\}$ contains $\mathcal{K}$ set of parameters, where $\mathcal{K}$ is the number of layers. A feedforward DNN structure with $\mathcal{K}$ layers is defined as a mapping $\mathcal{F}(\mathbf{u}_0, \theta) : \mathbb{R}^{N_0 \times 1} \mapsto \mathbb{R}^{N_K \times 1}$ of the input vector $\mathbf{u}_0 \in \mathbb{R}^{N_0 \times 1}$ to an output vector in $\mathbb{R}^{N_K \times 1}$ with the $\mathcal{K}$ iterative processing steps and expressed as

$$\mathbf{u}_k \triangleq f_k(\mathbf{u}_{k-1}; \theta_k), \quad k = 1, 2, \dots, \mathcal{K} \quad (11)$$

where $f_k(\mathbf{u}_{k-1}; \theta_k) : \mathbb{R}^{N_0 \times 1} \mapsto \mathbb{R}^{N_K \times 1}$ defines the mapping implemented by the k th DNN layer. This mapping depends on both a set of θ_k parameters and the $\mathbf{u}_{k-1}$ output vector from the

previous $(k-1)$ th layer. The mapping function can be a function of random variables. Hence, $f_k(\cdot; \cdot)$ may have a stochastic structure. Since all neurons in the proposed DNN architectures are fully connected to all following neurons, mapping function $f_k(\mathbf{u}_{k-1}; \theta_k)$ can be written via $\theta_k \triangleq \{\mathbf{W}_k, \mathbf{b}_k\}$ as follows:

$$f_k(\mathbf{u}_{k-1}; \theta_k) = \varsigma(\mathbf{W}_k \mathbf{u}_{k-1} + \mathbf{b}_k). \quad (12)$$

Here, $\mathbf{W}_k \in \mathbb{R}^{N_k \times (N_{k-1})}$ and $\mathbf{b}_k \in \mathbb{R}^{N_k \times 1}$ refer to the neurons' weight and bias vectors, respectively, and $\varsigma(\cdot)$ denotes an activation function. It should be stated that the basic mapping structure and processing steps are valid for both DNNs deployed in the system.

Proposed DNN architectures are shown in Fig. 5, where both DNNs consist of four hidden layers, one input, and one output layer, and each hidden layer has 256 neurons with a fully connected structure. In the proposed models, both rectified linear unit (ReLU) and hyperbolic tangent (tanh) activation functions are considered to be deployed at hidden layers. ReLU and tanh functions can be defined as

$$\varsigma_{\text{ReLU}}(z) = \max(0, z) \quad (13)$$

$$\varsigma_{\text{tanh}}(z) = \frac{\sinh z}{\cosh z} = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (14)$$

for a neuron with input z , respectively [22]. For both DNNs, we have applied ReLU and tanh activation functions consecutively to evaluate accuracy and convergence speed for the particular tasks in Table I. It is observed that ReLU converged nearly 5 times faster on average than tanh during the training phase, showing the same or better accuracy levels.

As the inputs for both DNNs in the model exhibit sequence data characteristics, both DNN's input layers are configured

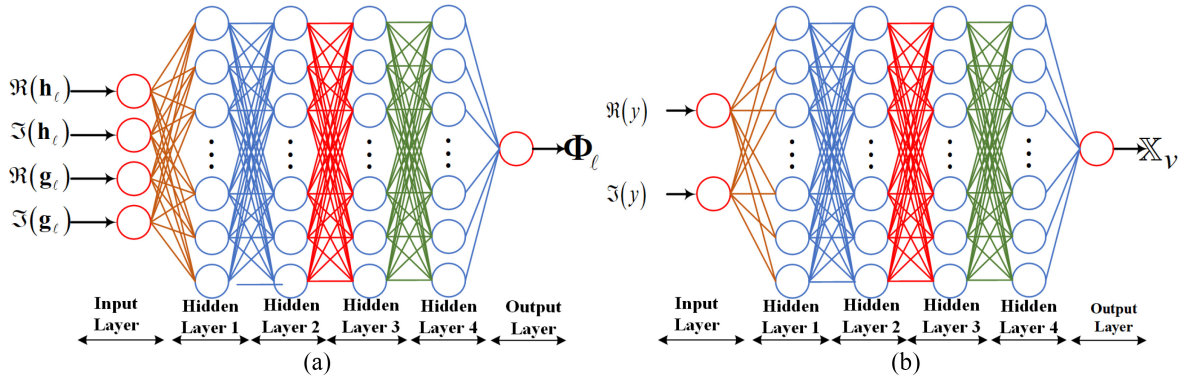


Fig. 5. Proposed DNN architectures (a) at the relay and (b) at the destination, for $\text{DNN}_R - \text{CRIS}/\text{DNN}_{R,D} - \text{CRIS}$ scheme.

TABLE I
RELU AND TANH TRAINING PERFORMANCE COMPARISON

	ReLU	tanh
Mini-batch Loss	0,0385	0,0391
Mini-batch RMSE	0,028	0,028
No. of Iterations	60	250
Time Elapsed	5'23"	25'35"

as the sequence input layer so that channel coefficients and received symbols can be used as input sequences for DNNs deployed at the relay and destination, respectively, albeit evaluated as different data types. On the other hand, the most significant difference between the two DNNs is the output layer implementation, which characterizes the network response behavior. The output layer of the DNN at R is configured as a regression layer, where the output will be complex phase adjustment estimations $\hat{\Phi}_\ell$. In contrast, the output layer of the DNN at D is configured as a classification layer for M -QAM modulated symbol estimations $\hat{x}$. In a typical classification network, a classification layer follows a softmax layer with the activation function defined as:

$$\zeta_{\text{softmax}}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}. \quad (15)$$

Here, $\mathbf{z}$ and z_i denote K vectors from previous layer and the elements of the input vector, respectively, where $i = 1, \dots, K$ and $\mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K$ [34]. The objective of using the softmax function is to convert K vectors at the input to K vectors at the output whose total is equal to 1. In this way, those vectors are normalized and mapped onto a probability distribution, and the vector values with the highest probability are transmitted to the next layer for classification.

Overfitting is a common problem in machine learning as the network performs too well on the training data set but performs poorly when it encounters new data. In the considered system, 10% of the training data set is split and reserved as validation data to evaluate the model during the training process in terms of generalization performance, with remaining data reserved for training.

B. Training Data Generation and the Training Process

For the proposed models, the preprocessing of the CSI data involves processing the CSI data before it is applied to DNN

as input, where the data is arranged and manipulated to be deployed as feature vectors and then applied to DNN at D . In this section, training data generation and training process are explained for $\text{DNN}_R - \text{CRIS}$ and $\text{DNN}_{R,D} - \text{CRIS}$ models.

1) *DNN_R - CRIS Model*: Training of the proposed DNN model requires feature vector samples, consisting of CSI data for incoming and outgoing channels $h_{\ell,n}$ and $g_{\ell,n}$, respectively, and adjusted phase angles $\phi_{\ell,n}$ accordingly. Because of the processing constraints of DNN, complex channel coefficients are separated into real and imaginary parts, as using the complex CSI data directly as input is not guaranteed to provide consistent results [35]. Preservation of the complete channel information is crucial, so the imaginary part should be processed along with real part, consequently, doubling the number of features per sample. Thus, the resulting feature vector set becomes as

$$\mathbf{F}_{\text{train}}^i = [\Re(h_{\ell,n}) \Im(h_{\ell,n}) \Re(g_{\ell,n}) \Im(g_{\ell,n})]_{1 \times 4}. \quad (16)$$

Here, $\mathbf{F}_{\text{train}} \triangleq \mathbf{F}_{\text{train}}^1, \mathbf{F}_{\text{train}}^2, \dots, \mathbf{F}_{\text{train}}^s$ for s samples, $i = 1, 2, \dots, s$ and $\mathbf{F}_{\text{train}}^i \in \mathbf{F}_{\text{train}}$. Likewise, for each feature vector, the corresponding phase vector $\phi_{\ell,n}$ is separated into real and imaginary parts, forming the output vector as

$$\mathbf{O}_{\text{train}}^i = [\Re(\phi_{\ell,n}) \Im(\phi_{\ell,n})]_{1 \times 2} \quad (17)$$

where $\mathbf{O}_{\text{train}} \triangleq \mathbf{O}_{\text{train}}^1, \mathbf{O}_{\text{train}}^2, \dots, \mathbf{O}_{\text{train}}^s$ for $i = 1, 2, \dots, s$ and $\mathbf{O}_{\text{train}}^i \in \mathbf{O}_{\text{train}}$. Finally, corresponding feature and output vector elements are concatenated to create the training data set as

$$D_{\text{train}} = \{ \{ \mathbf{F}_{\text{train}}^1, \mathbf{O}_{\text{train}}^1 \}, \{ \mathbf{F}_{\text{train}}^2, \mathbf{O}_{\text{train}}^2 \}, \dots, \{ \mathbf{F}_{\text{train}}^s, \mathbf{O}_{\text{train}}^s \} \}. \quad (18)$$

The objective of the training process of the proposed DNN algorithm is to optimize the weight ($\mathbf{W}_k$) and bias ($\mathbf{b}_k$) parameters of the DNN so that the difference between the actual output value of the DNN and the target value is minimized. To achieve this, the weights and bias vectors of the hidden layers of the DNN are updated at each iteration during training. $\mathbf{W}_k$ and $\mathbf{b}_k$ are the weight and bias vectors of the k th layer with $k = 1, 2, \dots, \mathcal{K}$, where $\mathcal{K} = 5$. According to the proposed model, DNN output $\hat{\Phi}_\ell$ for each relay is a function of $\mathbf{F}_{\text{train}}$ and $\Theta \triangleq \{\theta_1, \theta_2, \dots, \theta_s\}$ with $\theta_k \triangleq \{\mathbf{W}_k, \mathbf{b}_k\}$, where

Θ denotes the training parameter set consisting of $\mathbf{W}_k$ and $\mathbf{b}_k$ values. The difference between the actual output and target values is determined by the loss function, which is defined for the proposed DNN architecture as

$$\mathcal{L}(\Theta) = \frac{1}{2} \sum_{i=1}^s \left\| \phi_{\ell,n}^i - \hat{\phi}_{\ell,n}^i(\Theta) \right\|^2 \quad (19)$$

where $\phi_{\ell,n}$ is the target output value, $\hat{\phi}_{\ell,n}$ is the actual DNN output, and i is the index of the responses. On each iteration of the network, optimization algorithm constantly compares the loss value for each respective DNN output and optimizes $\mathbf{W}_k$ and $\mathbf{b}_k$ values on each layer to converge to the desired value. Optimization is carried out for each iteration based on the Adam (adaptive moment estimation optimization algorithm) optimizer, which can be defined as

$$\Theta_{\mu+1} = \Theta_{\mu} - \frac{\eta m_{\mu}}{\sqrt{v_{\mu}} + \epsilon} \quad (20)$$

where η is the learning rate determining the step size used to update the weights, ϵ is a smoothing term to avoid division by zero, and μ is the iteration index [36]. Adam optimizer updates the training parameter Θ at each iteration based on elementwise moving averages of both the parameter gradients and their squared values m_{μ} and v_{μ} , respectively, which can be expressed as

$$m_{\mu} = \delta_1 m_{\mu-1} + (1 - \delta_1) \nabla \mathcal{L}(\Theta) \quad (21)$$

$$v_{\mu} = \delta_2 m_{\mu-1} + (1 - \delta_2) [\nabla \mathcal{L}(\Theta)]^2. \quad (22)$$

Here, δ_1 and δ_2 are the decay rates of the moving averages, and $\nabla \mathcal{L}(\Theta)$ is the gradient of the loss function given in (19). If the gradients in (21) and (22) stay close after several iterations, the moving averages assist for the parameter updates to gain momentum in a certain direction. On the other hand, if the gradients contain a great amount of noise, then the parameter updates will be smaller due to the moving averages of the gradients getting smaller. This mechanism of the Adam is crucial for our case since the received signals contain a great deal of channel noise, exposed to path loss and occasionally mismatched phase angles at the relay [37].

2) *DNN_{R,D} – CRIS Model*: As the proposed DNN is supposed to be deployed at the destination, input data for the DNN will be noisy, path faded, and modulated signals reaching the destination (y_{ℓ}), and the output will be the estimations for original transmitted symbols ($\hat{x}$). The estimation problem for this case is a multiclass classification problem such that for every received signal y_{ℓ} , the DNN-based estimator should converge to the $\hat{x}$ estimate closest to the original symbol, which will be chosen from a predetermined symbol set.

The symbol set and the number of classes to be used in the classification process are based on the modulation order, such that, for the M th modulation order, there will be M symbols in the symbol set and M classes, consequently. In this respect, for the M th order, the symbol set and the respective classes can be defined as

$$\mathcal{S} = \{x_{S_1}, x_{S_2}, \dots, x_{S_M}\} \quad (23)$$

$$\mathcal{C} = \{\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_M\}. \quad (24)$$

Algorithm 1 Theoretical Training Algorithm for DNN @ R_{ℓ}

Input: CSI data - $h_{\ell,n}, g_{\ell,n}$

Output: Trained DNN at R_{ℓ} - $\mathbf{W}_k, \mathbf{b}_k$

Initialisation: Initialize the DNN parameters - $\mathbf{W}_k, \mathbf{b}_k$, and loss $\mathcal{L}(\Theta)$ are set to zero.

LOOP Process

1: **for** $i = 1$ to s **do**

2: Pre-process the CSI data - Real and imaginary parts of $h_{\ell,n}$ and $g_{\ell,n}$ are separated and rearranged as in (16), generating feature vector $\mathbf{F}_{\text{train}}^i$.

3: Compute the theoretical adjusted phase vector $\phi_{\ell,n}$ maximizing instantaneous SNR as in (5).

4: Pre-process the phase data found in step 3 - Real and imaginary parts of $\phi_{\ell,n}$ are separated and rearranged as in (17), generating output vector $\mathbf{O}_{\text{train}}^i$.

5: **end for**

6: Generate the training data - Feature and output vectors are concatenated to form D_{train} as in (18).

7: Train the DNN until $\mathcal{L}(\Theta)$ is minimized with respect to (19) and (20).

8: **return** Trained DNN at R_{ℓ}

Here, $\mathcal{C}$ is basically a set of class labels assigned for each member of the symbol set $\mathcal{S}$, where $x_{S_v} \in \mathcal{S}$ and $\mathbb{X}_v \in \mathcal{C}$ for $v = 1, 2, \dots, M$. It can be seen that each element of $\mathcal{C}$ can be mapped to the respective element in $\mathcal{S}$ and vice versa, which ultimately results in the estimation of $\hat{x}_s$ as seen in Fig. 4. This relation between the symbol set and classes can be represented by $\{x_{S_v} \longleftrightarrow \mathbb{X}_v\} \longrightarrow \hat{x}_s$. In this case, feature vector and the corresponding output vector for the proposed DNN can be stated as

$$\mathbf{F}_{\text{train}}^i = [\Re(\mathbf{y}_{\ell,n}), \Im(\mathbf{y}_{\ell,n})]_{1 \times 2} \quad (25)$$

$$\mathbf{O}_{\text{train}}^i = [\mathbb{X}_v^i]_{1 \times 1}, \quad v = 1, 2, \dots, M. \quad (26)$$

Here, it should be noted that $\mathbf{y}_{\ell,n}$ is separated into real and imaginary parts as in (16) for $i = 1, 2, \dots, s$, and $\mathbb{X}_v$ is a class category for $\mathbf{O}_{\text{train}}^i$. In this case, the training data set can be generated by concatenating feature vectors and corresponding symbol/class pairs for the sample space s , as shown in (18).

For the multiclass classification problem, the loss function which will be applied in the process can be defined as

$$\mathcal{L}(\Theta) = -\frac{1}{s} \sum_{i=1}^s \sum_{v=1}^M \mathbb{X}_v^i \ln(\hat{\mathbb{X}}_v^i(\Theta)) \quad (27)$$

where s is the number of samples, M is the number of classes, $\hat{\mathbb{X}}_v^i$ is the actual output value for the v th class at the i th sample, and $\mathbb{X}_v^i$ is the target value for the same sample. Here, $\ln(\cdot)$ denotes the natural logarithm function. As the classification layer follows the softmax layer, $\hat{\mathbb{X}}_v^i(\Theta)$ value will be the output of the softmax layer. Equation (27) is also called as the cross-entropy loss between the actual and target outputs [34].

For optimizing the loss function in (27), we also deploy Adam and follow the same blueprint stated for the previous DNN as in (20)–(22).

Based on the training processes presented so far, theoretical algorithms for both DNNs can be expressed as in Algorithms 1 and 2.

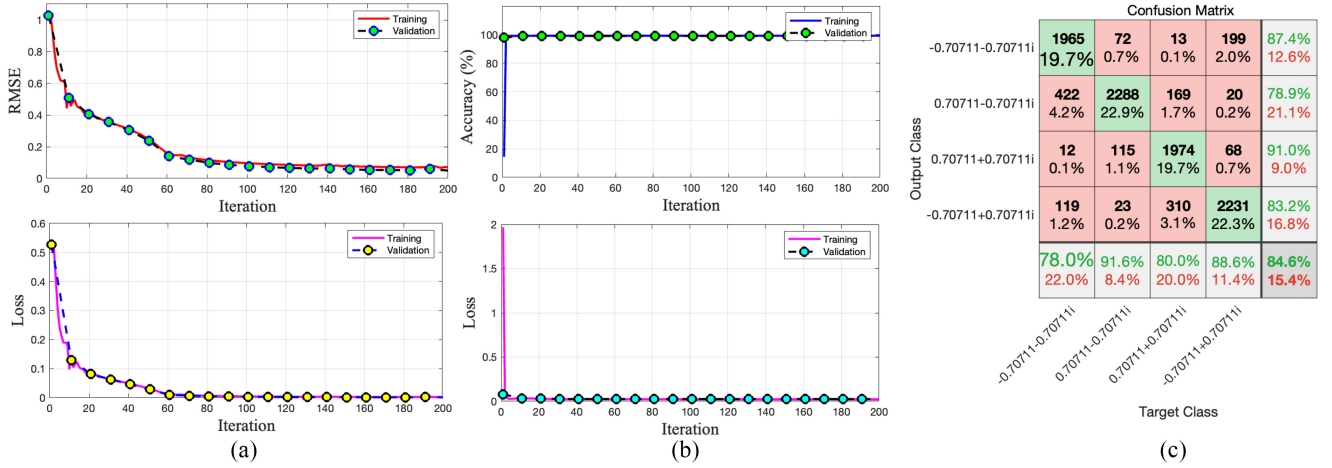


Fig. 6. Training progress of the DNN (a) at R_l , in terms of RMSE and loss using ReLU, (b) at D , in terms of accuracy (%) and loss using ReLU, and (c) confusion matrix for DNN at D with $M = 4$ for SNR = -40 dB.

Algorithm 2 Theoretical Training Algorithm for DNN @ D

Input: Original transmitted symbols (x) and noisy, path faded and modulated signals received at the destination ($y_{l,n}$).

Output: Trained DNN at D - W_k, b_k

Initialisation: Initialize the DNN parameters - W_k, b_k , and loss $\mathcal{L}(\Theta)$ are set to zero.

LOOP Process

- 1: **for** $i = 1$ to s **do**
- 2: Pre-process received signals at the destination - Real and imaginary parts of $y_{l,n}$ are separated and rearranged as in (25), generating feature vector $\mathbf{F}_{\text{train}}^i$.
- 3: Extract the transmitted symbols and convert symbol set into a class set as in (23) and (24), respectively.
- 4: Generate output vectors - For every feature vector, the corresponding class is designated as an output vector $\mathbf{O}_{\text{train}}^i$ as in (26).
- 5: **end for**
- 6: Generate the training data - Feature and output vectors are concatenated to form D_{train} as in (18).
- 7: Train the DNN until $\mathcal{L}(\Theta)$ is minimized with respect to (27) and (20).
- 8: **return** Trained DNN at D

IV. NUMERICAL RESULTS AND DISCUSSIONS

A. Simulation Parameters for DNN_R - CRIS/DNN_{R,D} - CRIS

Simulation is based on the channel and system model described in Section II, where S , R_l , and D have been positioned as depicted in Fig. 3. All relays are kept within the half-circle area by choosing $\pi/2 < \theta_l < \pi$, with d_{SD} , d_{SR_l} , and d_{R_lD} distances normalized with respect to $d_{SD} = 1$. The SNR used in the simulations herein is defined as $\text{SNR}(\text{dB}) = 10 \log_{10}(E_s/N_0)$. The number of reflecting elements on RIS-based relays is originally specified between 8 and 128 for observing its effect on relay configurations and BER performances. However, due to space constraints, plots for 64 and 128 reflectors have not been included in the manuscript as the results do not show a remarkable deviation from the overall trend. All channels are subject to Rayleigh fading and path loss effects with path loss exponent c chosen as 4 for Figs. 8–11, and $c = 3$ for Fig. 12 [30].

TABLE II
TRAINING PARAMETERS

Training Parameter	DNN@Relay	DNN@Destination
Modulation Order (M)		4/8
Num. of Transmit Antennas		1
Num. of Reflect. Elements (N)		8/32
Pathloss Exponent (c)		4
Num. of Hidden Layers		4
Batch Size		256
Validation Split		10%
Num. of Samples (s)	360000	45000
Iteration Steps	300	150
Learning Rate (η)	0.003	0.003

We use MATLAB and its toolboxes for the simulation with the basic training parameters given in Table II. As stated in Section III-A, ReLU activation function is opted for deployment in hidden layers for both DNN models in all simulation phases. During training, 10% of the training data set is reserved for validation by the validation split, and validation frequency is set as 10 for both DNNs. Another critical parameter for the training phase is the batch size, which determines the size of the data batch—also called as a mini-batch—processed by the network on each iteration. Batch size is basically a subset of the training data set, and for both DNN models, it is set as 256.

B. DNN Performance Analysis

Performances of the DNNs deployed in the proposed model have been evaluated using a number of tools during training and transmission phases.

Fig. 6(a) and (b) shows the training progress plots for DNNs at R and D , respectively, providing information on the training and validation performance in terms of root mean square error (RMSE) and loss values for DNN at R , and accuracy percentage for DNN at D . These metrics are calculated during training for each mini-batch, processed by the network on each iteration. As it can be observed from the training and validation curves, both DNNs show a good generalization performance with little to no need for extra measures for overfitting.

Loss values for the proposed DNNs at R and D can be directly computed by the functions defined in (19) and (27), respectively. On the other hand, RMSE for a typical regression network–DNN at R_ℓ in our case, can be defined as

$$\text{RMSE} = \sqrt{\frac{1}{s} \sum_{i=1}^s (t_i - \hat{t}_i)^2}. \quad (28)$$

Here, t_i is the target output, $\hat{t}_i$ is the actual output, and s denotes the number of samples or responses in general terms [38]. RMSE and loss curves in a regression network will follow a similar path, as loss function is half mean-squared error computed by the network for optimizing the parameters.

The accuracy of the DNN at D can also be monitored during the transmission phase, using confusion matrices. The confusion matrix for classification networks shows the percentage of the actual DNN outputs matching with the target outputs for each prediction cycle of the network. Fig. 6(c) shows the confusion matrix for DNN deployed at D for $M = 4$ and $\text{SNR} = -40$ dB. Here, the horizontal and vertical axes show the target output and actual output, respectively. The cells on the green diagonal show the percentage of the successfully matched symbols for each symbol class, and the cell at the lower right corner shows the overall percentage of accuracy for all symbol classes. During simulation, the DNN output accuracy has a direct impact on the BER performance, such that a high estimation accuracy for symbol detection improves BER values considerably.

During the simulation, both DNNs at the relay and destination have been trained for a range of SNR levels for optimum accuracy, and we observed the generalization ability of the algorithm when the training and transmission SNRs were different. It is found that the general accuracy level improves with the increase in SNR as expected and has similar results for different training and transmission SNRs.

C. Complexity Analysis

In this section, we will analyze and compare the computational complexities of RS, MRC, and ML-based receivers against the DNN-assisted symbol detection algorithm deployed in the $\text{DNN}_{R,D}$ – CRIS model.

The complexity analysis for the proposed DNN model is based on the number of operations in each layer. As the input neurons function as the entry point for the signals and do not perform any operation, they will not be considered in the analysis as a layer.

In each neuron, a mapping function is performed as in (12), which involves the linear combination of the input with neuron parameters $\mathbf{W}_k$, $\mathbf{b}_k$, and later processed by the activation function. In a multilayer neural network, the complexity depends on the number of layers and the size of each layer, both of which scale the number of vector multiplications. Hence, for a multilayer neural network with $\mathcal{K}$ hidden layers and p inputs, if the k th hidden layer has n_k neurons with o neurons at the output, the total number of multiplications will be

$$O(pn_1 + on_{\mathcal{K}} + \sum_{k=1}^{\mathcal{K}-1} n_k n_{k+1}) \quad (29)$$

where all the layers are assumed to be fully connected.

TABLE III
COMPUTATIONAL COMPLEXITY ANALYSIS

RS	MRC	ML	DNN
$O(N + 4M)$	$O(L(N + 1) + 4M)$	$O(LN + 4L^2M)$	$O(pn_1 + n_{\mathcal{K}}o + \sum_{k=1}^{\mathcal{K}-1} n_k n_{k+1})$

TABLE IV
PARAMETERS OF COMPLEXITY SCENARIOS AND DNN CONFIGURATIONS

	Scenario 1	Scenario 2	Scenario 3
Num. of Relays (L)	4	6	24
Modulation Order (M)	4	8	16
Num. of Refl. Elements (N)	16	32	128
DNN-1	$\mathcal{K} = 4, n_1 = n_2 = n_3 = n_4 = 256$		
DNN-2	$\mathcal{K} = 2, n_1 = n_2 = 16$		
DNN-3	$\mathcal{K} = 2, n_1 = n_2 = 8$		

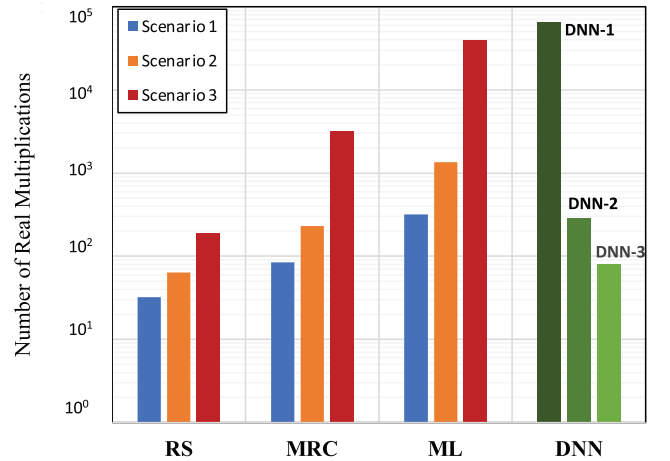


Fig. 7. Complexity comparisons of RS, MRC, ML, and DNN-based CRIS systems.

The computational complexities of RS, MRC, ML, and DNN-based systems in terms of number of real multiplications are illustrated in Table III. For RS, MRC, and ML-based detectors, we evaluated three scenarios with various L , M , and N values as given in Table IV and compared them against the DNN-based detectors with different layer and neuron configurations. It can be observed from Fig. 7 that the complexities of RS, MRC, and ML-based detectors increase as the L , M , and N usage grow in a large-scale network. In this regard, ML and RS are found to be the most and least computationally complex detection schemes, respectively, where the relay usage is the most decisive element for both. On the DNN side, the complexity depends solely on the layer and neuron configurations. Various DNN configurations are evaluated for comparison as given in Table IV, all of which exhibit nearly identical detection accuracy during simulations. It is found that the complexity can be reduced successfully to a minimum with DNN-3 configuration, where the complexity is lower than all ML and MRC scenarios and even lower than RS scenarios with a high number of network components.

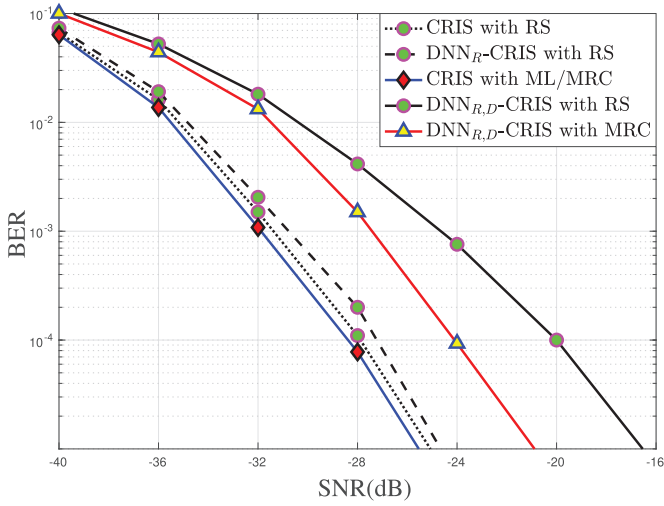


Fig. 8. BER comparisons of DNN_R - CRIS with RS, $DNN_{R,D}$ - CRIS with RS/MRC, and CRIS with RS/ML/MRC systems for $M = 4$, $L = 2$, and $N_1 = N_2 = 8$.

D. BER Analysis of DNN_R - CRIS/ $DNN_{R,D}$ - CRIS Schemes

In this section, we will evaluate the BER performances of the DNN_R - CRIS and $DNN_{R,D}$ - CRIS models using various relay and receiver configurations, including non-DNN RIS-based relay configurations (CRIS) with RS, MRC, and ML-based receivers at D as a benchmark for all BER performance comparisons. For all the models with RS scheme, only the best performing relay is depicted in the figures.

We start evaluating the BER performance of a two-relay cooperative system, using a 4-QAM modulation scheme with RS. Each RIS-based relay contains eight reflecting elements with relay distances $d_{SR_1} = 0.2$, $d_{R_1D} = 0.98$ and $d_{SR_2} = 0.5$, $d_{R_2D} = 0.86$ for $\theta_1 = \theta_2 = \pi/2$. In this configuration, R_1 is selected as the best performing relay according to (6), which is also the closest relay to S , and its BER performance is depicted in Fig. 8. It can be seen that performance of the DNN_R - CRIS model is very close to CRIS. The addition of DNN-assisted symbol estimation into the system ($DNN_{R,D}$ - CRIS) introduces a slight impact on the BER performance as expected, such that, for a BER level of 10^{-3} the SNR loss is about 6 dB. It is also noteworthy that the $DNN_{R,D}$ - CRIS with MRC shows comparable performance with CRIS with MRC system, even outperforming non-DNN benchmark configurations.

In Fig. 9, the number of reflecting elements is increased to 32, with all remaining parameters retained from the previous scenario. The immediate effect of increasing reflecting elements is observed as an SNR improvement of about 7 dB for $N = 32$ at the BER of 10^{-3} or below. We see that configurations with the DNN_R - CRIS model show nearly identical performance with non-DNN benchmark configurations, and the impact of DNN-based destination on SNR remains at similar levels. Increasing the number of reflecting elements on the relays improves the BER performances of DNN-based systems entirely, although preserving their performance characteristics.

Fig. 10 shows an RS outcome among a four-relay system with relay distances of $d_{SR_1} = 0.4$, $d_{R_1D} = 0.91$, $d_{SR_2} = 0.55$,

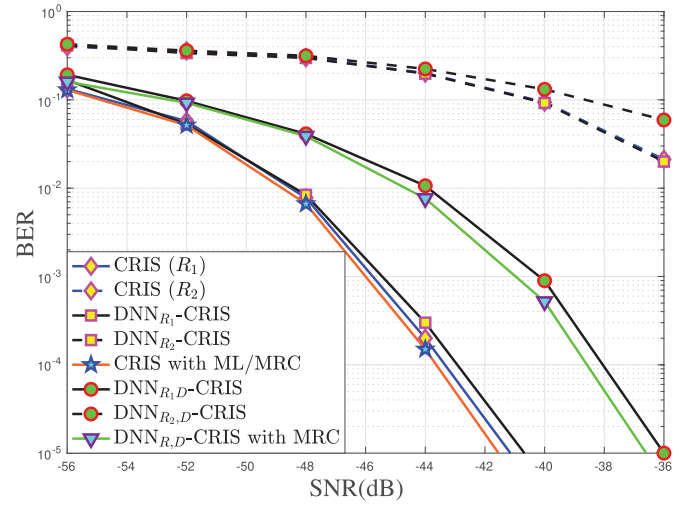


Fig. 9. BER comparisons of DNN_R - CRIS/ $DNN_{R,D}$ - CRIS and CRIS systems for $M = 4$, $L = 2$, and $N_1 = N_2 = 32$.

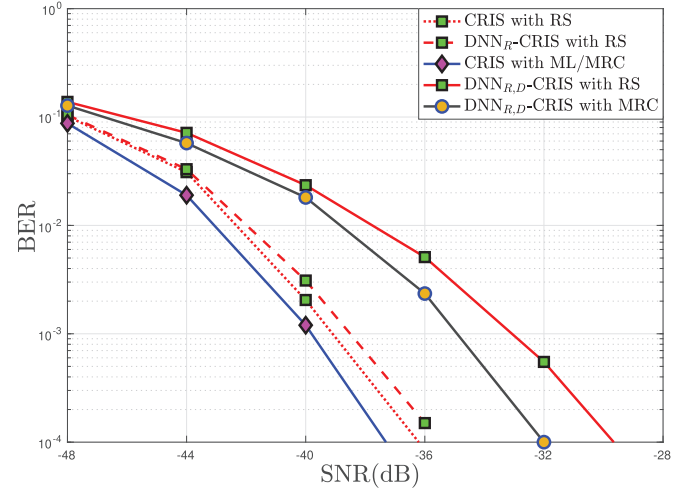


Fig. 10. BER comparisons of DNN_R - CRIS with RS, $DNN_{R,D}$ - CRIS with RS/MRC, and CRIS with RS/ML/MRC systems for $M = 4$, $L = 4$, and $N_1 = N_2 = N_3 = N_4 = 8$.

$d_{R_2D} = 0.83$, $d_{SR_3} = 0.65$, $d_{R_3D} = 0.76$, $d_{SR_4} = 0.2$, and $d_{R_4D} = 0.98$, each R having eight reflecting elements. Here, as in Fig. 8, the best-performing relay is the closest one to S , which is R_4 in this case. Once more, we observe that the DNN_R - CRIS scenario shows nearly identical BER performance with non-DNN configuration as in the previous cases. Here, the path loss effects become clearer as the relays are distributed throughout the model layout, such that configurations incorporating relays closer to S or D show a better BER performance as the signals are exposed to a lesser attenuation. The most significant outcome here is that the BER value is found to be maximum when $d_{SR_\ell} = d_{R_\ell D}$.

We also compared the BER performances for $M = 4$ and $M = 8$ schemes as shown in Fig. 11, using RS among the relays with distances $d_{SR_1} = 0.9$, $d_{R_1D} = 0.43$, $d_{SR_2} = 0.35$, and $d_{R_2D} = 0.93$. It can be seen that, for both modulation schemes, the performance characteristics remain mostly unchanged as the best performance is achieved for R_1 , the closest one to D , where the DNN-based destination introduced a certain amount of SNR loss. The only remarkable outcome is the general performance setback for $M = 8$ as expected,

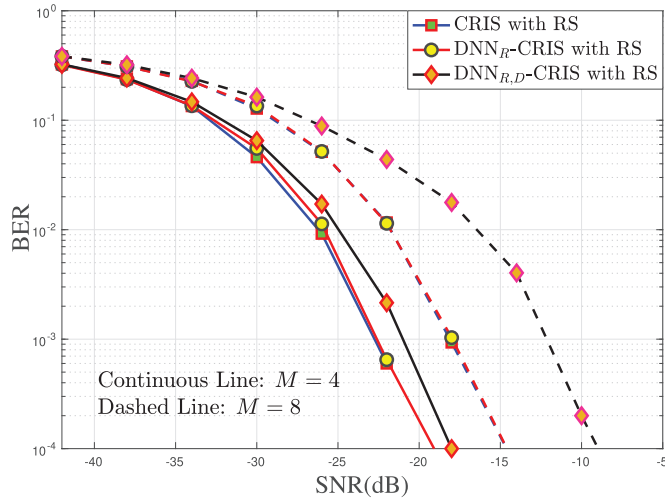


Fig. 11. BER comparisons of DNN_R - CRIS with RS, $\text{DNN}_{R,D}$ - CRIS with RS, and CRIS with RS systems for $M = 4, 8, L = 2$, and $N_1 = N_2 = 8$.

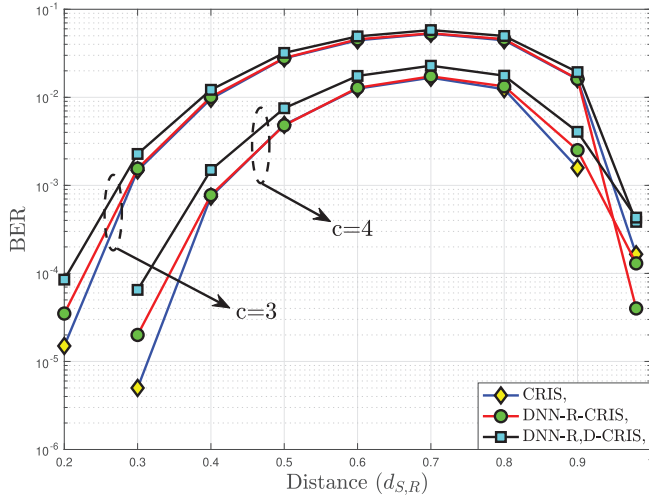


Fig. 12. BER versus d_{SR} comparisons of DNN_R - CRIS/ $\text{DNN}_{R,D}$ - CRIS and CRIS systems with a single relay for $M = 4, N = 8, \text{SNR} = -20$ when $c = 3, 4$.

such that at the BER of 10^{-3} , the SNR loss is about 5 dB for DNN_R - CRIS and 8 dB for $\text{DNN}_{R,D}$ - CRIS models.

Finally, in Fig. 12, we can observe the effect of distance on the BER performance clearly, as the worst performance is shown at $d_{SR} = 0.7$ for all configurations, which is close to midpoint between S and R . Furthermore, the effect of path loss exponent (c) can also be observed on the BER performance, such that increasing c results in a better BER performance. This outcome results from the fact that all the distance values are normalized with respect to $d_{SD} = 1$, where $0 < d_{SR_\ell}, d_{R_\ell} \leq 1$.

V. CONCLUSION AND FUTURE WORKS

In this work, we realize a DL-assisted RIS-based cooperative communication system with a DNN-based symbol detector at the destination, where all the RIS elements are considered to be interconnected through an IoT network. The effect of the path loss on the DNN_R - CRIS and $\text{DNN}_{R,D}$ - CRIS

models is demonstrated in terms of BER performance. We incorporate DNN performance analysis techniques during the training and/or transmission phases for evaluating the individual DNN performances and analyze the BER performances to evaluate the total system performance in all deployment scenarios. The simulation results show that DNN-assisted RIS-based relays can effectively maximize the reflected signals and exhibit satisfactory performance against the non-DNN benchmark configurations in general. Combining DNN-assisted relays with DNN-based symbol detectors at the destination also shows acceptable performance with low system complexity, especially in conjunction with RIS-based relays that harbor large numbers of reflectors. Implementation flexibility is another benefit worth mentioning; DNN-based models can easily be scaled down for a reduced system complexity with nearly identical BER performance, where it will be possible to use single or double DNN configurations for different channel and network conditions. Moreover, using MRC at the destination to combine multirelay signals for the dual DNN scenarios yields promising results, and applying RS schemes to the final model is also possible.

In the context of this work, it is assumed that the CSI is perfectly known at the relay. It will be a broad and challenging field of study to consider similar scenarios under totally blind or imperfect CSI conditions. The performance of similar systems with imperfect CSI are already discussed in [10], [22], and [29], and we expect to extend some of the methods proposed in these works in our future studies where more advanced DL techniques and DNN architectures will presumably be needed.

REFERENCES

- [1] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.
- [2] B. Coll-Perales, J. Gozalvez, and J. L. Maestre, "5G and beyond: Smart devices as part of the network fabric," *IEEE Netw.*, vol. 33, no. 4, pp. 170–177, Jul./Aug. 2019.
- [3] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M.-S. Alouini, and R. Zhang, "Wireless communications through reconfigurable intelligent surfaces," *IEEE Access*, vol. 7, pp. 116753–116773, 2019.
- [4] E. Basar and H. V. Poor, "Present and future of reconfigurable intelligent surface-empowered communications [perspectives]," *IEEE Signal Process. Mag.*, vol. 38, no. 6, pp. 146–152, Nov. 2021, doi: [10.1109/MSP.2021.3106230](https://doi.org/10.1109/MSP.2021.3106230).
- [5] L. Subrt and P. Pechac, "Controlling propagation environments using intelligent walls," in *Proc. 6th Eur. Conf. Antennas Propag. (EUCAP)*, 2012, pp. 1–5.
- [6] S. Hu, F. Rusek, and O. Edfors, "Beyond massive MIMO: The potential of data transmission with large intelligent surfaces," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2746–2758, May 2018.
- [7] C. Huang, G. C. Alexandropoulos, A. Zappone, M. Debbah, and C. Yuen, "Energy efficient multi-user MISO communication using low resolution large intelligent surfaces," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–6.
- [8] Q. Wu and R. Zhang, "Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 106–112, Jan. 2020.
- [9] C. Huang, A. Zappone, M. Debbah, and C. Yuen, "Achievable rate Maximization by passive intelligent mirrors," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2018, pp. 3714–3718.
- [10] E. Basar, "Transmission through large intelligent surfaces: A new frontier in wireless communications," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, 2019, pp. 112–117.

- [11] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network: Joint active and passive Beamforming design," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.
- [12] O. Salan, F. Bayar, and H. Ilhan, "Performance evaluation of RIS-based SSK and SM schemes with perfect and imperfect channel phase estimation over Weibull fading channels," *AEU Int. J. Electron. Commun.*, vol. 136, Jul. 2021, Art. no. 153713. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1434841121001102>
- [13] B. Onal, F. Cogen, E. Aydin, and N. Kabaoglu, "Reconfigurable intelligent surface empowered differential chaos shift keying assisted media-based modulation over Nakagami- m fading channels," in *Proc. Int. Conf. Elect. Electron. Eng.*, 2021, pp. 1–5.
- [14] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "A new wireless communication paradigm through software-controlled metasurfaces," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 162–169, Sep. 2018.
- [15] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "Realizing wireless communication through software-defined hypersurface environments," in *Proc. IEEE 19th Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, 2018, pp. 14–15.
- [16] E. Aydin and H. Ilhan, "SNR-based relay selection scheme for cooperative relay networks," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, 2015, pp. 448–453.
- [17] E. Aydin, E. Basar, H. Ilhan, and N. Kabaoglu, "C-MRC-Based cooperative spatial modulation with antenna selection," *Int. J. Commun. Syst.*, vol. 33, no. 17, 2020, Art. no. e4600. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4600>
- [18] F. Cogen and E. Aydin, "Cooperative quadrature spatial modulation with euclidean distance and capacity optimized antenna selection," *Int. J. Commun. Syst.*, vol. 34, no. 1, 2021, Art. no. e4663. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4663>
- [19] M. Di Renzo et al., "Reconfigurable intelligent surfaces vs. relaying: Differences, similarities, and performance comparison," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 798–807, 2020.
- [20] E. Björnson, O. Özdogan, and E. G. Larsson, "Intelligent reflecting surface versus decode-and-forward: How large surfaces are needed to beat relaying?" *IEEE Wireless Commun. Lett.*, vol. 9, no. 2, pp. 244–248, Feb. 2020.
- [21] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4157–4170, Aug. 2019.
- [22] S. Khan, S. Y. K. Shin, N. Haider, and S. Y. Shin, "Deep-learning-aided detection for reconfigurable intelligent surfaces," 2020, *arXiv:1910.09136*.
- [23] B. Akdemir, M. A. Karabulut, and H. Ilhan, "Performance of deep learning methods in DF based cooperative communication systems," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, 2021, pp. 1–6.
- [24] C. Huang, G. C. Alexandropoulos, C. Yuen, and M. Debbah, "Indoor signal focusing with deep learning designed reconfigurable intelligent surfaces," 2019, *arXiv:1905.07726*.
- [25] X. Ying, U. Demirhan, and A. Alkhateeb, "Relay aided intelligent reconfigurable surfaces: Achieving the potential without so many antennas," 2020, *arXiv:2006.06644*.
- [26] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 375–388, Jan. 2021.
- [27] Z. Wang, L. Liu, and S. Cui, "Channel estimation for intelligent reflecting surface assisted multiuser communications: Framework, algorithms, and analysis," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6607–6620, Oct. 2020.
- [28] L. Wei, C. Huang, G. C. Alexandropoulos, C. Yuen, Z. Zhang, and M. Debbah, "Channel estimation for RIS-empowered multi-user MISO wireless communications," *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 4144–4157, Jun. 2021.
- [29] Q. Chen, S. Zhang, S. Xu, and S. Cao, "Efficient MIMO detection with imperfect channel knowledge—A deep learning approach," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2019, pp. 1–6.
- [30] T. Rappaport and L. Milstein, "Effects of radio propagation path loss on DS-CDMA cellular frequency reuse efficiency for the reverse channel," *IEEE Trans. Veh. Technol.*, vol. 41, no. 3, pp. 231–242, Aug. 1992.
- [31] J. Proakis and M. Salehi, *Digital Communications 5th Edition*. New York, NY, USA: McGraw Hill, 2007.
- [32] K. J. R. Liu, A. K. Sadek, W. Su, and A. Kwasinski, *Cooperative Communications and Networking*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
- [33] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available: <https://doi.org/10.1038/nature14539>
- [34] J. S. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2, 1989, pp. 211–217.
- [35] MATLAB, *Train Network With Complex-Valued Data*. Natick, MA, USA: MathWorks Inc., 2022. [Online]. Available: <https://www.mathworks.com/help/deeplearning/ug/train-network-with-complex-valued-data.html>
- [36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*.
- [37] Y. Song, M. R. A. Khandaker, F. Tariq, K.-K. Wong, and A. Toding, "Truly intelligent reflecting surface-aided secure communication using deep learning," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC2021-Spring)*, 2021, pp. 1–6.
- [38] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Hoboken, NJ, USA: Prentice-Hall, 1997.



Bulent Sagir was born in Istanbul, Turkey. He received the B.Sc. degree in electronics engineering from Istanbul University, Istanbul, in 1998, and the M.Sc. degree in electronics and communication engineering from Yildiz Technical University, Istanbul, in 2001, where he is currently pursuing the Ph.D. degree in electronics and communication engineering.

He is also currently with Türk Telekomünikasyon A.Ş., Istanbul, where he has served in engineering, project management, presales consultation, and technical management positions. His general research interests are deep learning, wireless communications, and information networks, with specific areas, including deep neural networks, unsupervised learning, DL-assisted signal and image processing applications, IoT networks, and cooperative diversity techniques.



Erdogan Aydin was born in Turkey. He received the B.S. (with highest Hons.) and M.S. degrees from Istanbul University, Istanbul, Turkey, in 2007 and 2010, respectively, and the Ph.D. degree from Yildiz Technical University, Istanbul, in 2016.

He is currently an Associate Professor with the Department of Electronics and Communication Engineering, Istanbul Medeniyet University, Istanbul. His primary research interests include MIMO systems, cooperative communication and diversity, index modulation, spatial modulation, media-based modulation, hexagonal constellations, nonorthogonal multiple access, visible light communication, chaos communication, statistical signal processing, and estimation theory.

Dr. Aydin has received the Best Paper Award including one from the IEEE International Conference on Communications 2018. He has served as a TPC member for several IEEE conferences.



Hacı İlhan (Senior Member, IEEE) received the B.Sc. degree in electronics and communication engineering from Yildiz Technical University, Istanbul, Turkey, in 1999, and the M.Sc. and Ph.D. degrees in electronics and communication engineering from Istanbul Technical University, Istanbul, in 2003 and 2011, respectively.

From 2001 to 2011, he was a Research and Teaching Assistant with the Communication Department, Istanbul Technical University. Since 2011, he has been with the Department of Electronics and Communication, Yildiz Technical University, where he is a Professor. His general research interests lie in communications theory and signal processing for communications with particular emphasis on wireless applications. Specific research areas include cooperative communication, MIMO communication techniques, space-time coding, multicarrier communication, network coding, spatial modulation, index modulation, full-duplex communication, NOMA, energy harvesting, SWIPT, vehicular communication, and MAC protocols for vehicular ad hoc networks.