Deep Learning Based Physical-Layer Receiver With Pre-denoise

Yunqi Feng, Yunxian Chen, Huimei Han, Weidang Lu, *Senior Member, IEEE*, Nan Zhao, *Senior Member, IEEE*, Arumugam Nallanathan, *Fellow, IEEE,* and Xiaoniu Yang

***Abstract*—Traditional physical-layer receivers are typically composed of step-by-step serial processing modules heavily relying on complex hardware implementations and algorithms, where the optimal global performance cannot be guaranteed. In addition, system noise degrades the reliability of each module, resulting in the cumulative effect of errors. In this correspondence, we proposes a novel intelligent receiver consisting of a pre-denoised network and a stacked long-short term memory (LSTM) network to achieve more reliable information recovery. Specifically, the residual convolutional neural network (CNN) is employed to extract time-frequency characteristics of noise to denoise preprocessing, and then we utilize the stacked LSTM network to recover the information. Simulation results demonstrate that the proposed scheme can archive a bit error rate (BER) performance close to the ideal maximum likelihood algorithm and outperform the baselines.**

***Index Terms*—Wireless communications, signal denoising, deep learning, convolutional neural network, stacked LSTM.**

I. INTRODUCTION

THE increasing development of wireless communications has facilitated the emergence of advanced wireless services such as autonomous driving and Internet of Things [1]. As a result, there are increasing demands for higher levels of reliability to support these services. However, the performance of wireless communication systems may be significantly degraded by various non-ideal factors, such as noise, in-phase (I) and quadrature-phase (Q) imbalances [2]. Thus, the design of physical-layer (PL) receiver is critical to ensure information recovery. To satisfy the requirement for more reliable information transmission, traditional PL receivers typically employ a step-by-step serial signal processing approach that includes channel estimation, equalization, demodulation, and decoding.

However, since the performance of subsequent modules may be impacted by the residual error of pre-processing module, the optimal performance of the whole system cannot be guaranteed due to the cumulative effect of errors [3]. Furthermore, although most signal processing algorithms in wireless communications are based on solid statistical and information-theoretic foundations, they often follow theoretical assumptions that do not fully reflect the practical environment [4]. In addition, the implementation of traditional PL receivers requires dedicated hardware modules, resulting in a higher cost and a significant amount of time and resource for development.

To address the limitations of traditional PL receivers, deep learning (DL) based intelligent receivers have been developed to improve the reliability and effectiveness of information recovery under complex wireless environments [5]. With the increasing computing power of hardware, DL is becoming a powerful tool in solving complex tasks [6]. Recently, there has been a surge in intelligent communication technologies that combine DL with signal processing, including modulation identification [7], channel estimation and equalization [8], [9], signal demodulation and channel decoding [10], [11]. The above studies only focus on the optimization of a specific signal processing module. In [12], Zheng *et al*. proposed a DeepReceiver that replaces the entire information recovery in an end-to-end manner, ensuring the overall optimal performance of the receiver. However, the highly complex network structure of DeepReceiver may result in a long training time and significant computational power requirements.

Moreover, since the presence of noise in communication systems will deteriorate the quality of the received signal, several DL-based noise suppression schemes have been proposed to enhance the information recovery. Lee *et al*. in [13] designed a one-dimensional denoising-based autoencoder for signal recovery with noise learning and suppression. Similarly, Yang *et al*. developed an intelligent denoiser that effectively diminishes noise and restore one-dimensional bit streams [14]. However, it should be noted that both the two denoisers are primarily designed to eliminate the correlated noise, where the correlation can be utilized to simplify the model.

Inspired by above discussions, we propose a novel intelligent receiver that comprises of a pre-denoising (PDN) module and a stacked LSTM network (SLSTM) module to recover the signal with high reliability. The PDN module is designed to extract the time-frequency characteristics of received signal for efficient denoising with a serial cascade of data processing and residual denoising (RDN) components. The subsequent SLSTM module is capable of recovering the transmitted bits from the denoised signal. Simulation results demonstrate that the PDN-SLSTM receiver achieves reliable information recovery and effectively reduces the noise effects, thereby improving the bit error rate (BER) performance.

The remainder of this paper is organized as follows. The system model of wireless communication systems is introduced in Section II. Section III discusses the proposed PDN-SLSTM receiver in detail. Simulation results are presented in Section IV, while the conclusions are derived in Section V.

Yunqi Feng, Yunxian Chen, Huimei Han, Weidang Lu and Xiaoniu Yang are with College of Information Engineering, Key Laboratory of Communication Networks and Applications of Zhejiang Province, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: yqfeng@zjut.edu.cn, 2112103028@zjut.edu.cn, hmhan1215@zjut.edu.cn, luweid@zjut.edu.cn, yxn2117@126.com).

Nan Zhao is with the School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, P. R. China (e-mail: zhaonan@dlut.edu.cn).

Arumugam Nallanathan is with School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS, U.K. (e-mail: a.nallanathan@qmul.ac.uk).

II. System Model

Wireless communication systems typically consist of a transmitter and a receiver, as illustrated in Fig. 1. At the transmitter, the channel coding encodes the m-bit information bit stream $\boldsymbol{s} = [s_1, ..., s_m]^T$ into the n-bit codeword $\boldsymbol{c}$. Then, the codeword $\boldsymbol{c}$ is converted into a symbol $\boldsymbol{x}$ via modulation. The symbol $\boldsymbol{x}$ is then transmitted through the wireless channel, and the received symbol can be represented as

$$\boldsymbol{y} = \boldsymbol{h}\boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{h}$ denotes the multiplicative fading factor that accounts for multipath propagation, fading, interference, and other channel effects. Here, $\boldsymbol{n} \sim \mathcal{CN}(0, \sigma_n^2)$ is an additive white Gaussian noise (AWGN) vector, which is characterized by a complex Gaussian distribution with a mean of zero and a variance of σ_n^2.

The receiver plays a vital role in ensuring communication performance by accurately recovering the information from distorted signals. In order to recovery the original information, the demodulation and channel decoding modules are directly employed by the traditional PL receivers. In contrast, the PDN-SLSTM receiver is proposed as an alternative approach, depicted in the lower part of Fig. 1. It utilizes two deep neural network modules to replace the traditional information recovery modules. First, the PDN module denoises the decomposed I-Q components (i.e., Re($\boldsymbol{y}$) and Im($\boldsymbol{y}$)) of $\boldsymbol{y}$. Then, the denoised modulation symbol $\hat{\boldsymbol{x}}$ is further processed by the subsequent SLSTM module to estimate the information bit stream, obtaining $\hat{\boldsymbol{s}} = [\hat{s}_1, ..., \hat{s}_m]^T$.

III. The Proposed PDN-SLSTM Receiver

This section presents the model structure of PDN module, followed by the network architecture design of RDN module and SLSTM module.

A. Model Structure of PDN Module

The PDN module comprises three main components: short-time Fourier transform (STFT), RDN module, and short-time Fourier inverse transform (ISTFT). Unlike traditional denoising methods that only focus on the temporal characteristics of noise, the propose PDN module considers both time and frequency characteristics of the signal and noise. By utilizing the STFT with window size of l, we process the received signal and obtain a two-dimensional representation as

$$\boldsymbol{Y}[r,q] = \sum_{p=q}^{q+l-1} \mathrm{y}[p]\mathrm{w}[p-q+1]e^{-j\frac{2\pi p}{l}r} \in \mathbb{C}^{r\times q}, \tag{2}$$

where w represents the window function, and r and q denote the frequency and time domain intervals, respectively. Here, p is utilized to control the position of window function.

By separately performing the time-frequency transform on the received modulated symbols and the transmitted modulated symbols, the time-frequency domain noise $\boldsymbol{N}$ can be extracted. Traditional denoising methods only focus on learning a mapping function that predicts the denoising result without taking into account the time-frequency characteristics of the noise. In contrast, the proposed RDN module uses a residual learning

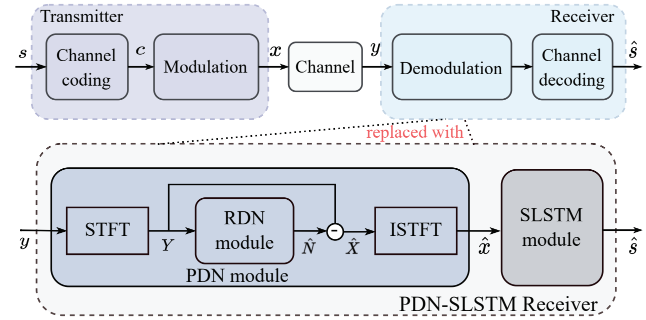

Fig. 1. Wireless communication systems and the PDN-SLSTM receiver.

method to train the residual mapping $\mathcal{H}(\boldsymbol{Y}) = \hat{\boldsymbol{N}}$ to predict the noise. As a result, the denoised modulation symbols can be attained in the time-frequency domain, i.e., $\hat{\boldsymbol{X}} = \boldsymbol{Y} - \hat{\boldsymbol{N}}$. Finally, $\hat{\boldsymbol{x}}$ can be recovered by performing ISTFT.

B. Network Architecture of RDN module

The performance of a neural network model is significantly impacted by its training process, which encompasses both the design of the network architecture and the acquisition of network parameters from the training data. In the network architecture design, convolutional neural networks (CNN) have been chosen due to their advantages including sparse connections, weight sharing and superior performance in two-dimensional signal processing. As shown in Fig 2, the network architecture of the proposed RDN module uses CNN in combination with a residual learning block.

The RDN module consists of three main layers: the convolution (Conv) layer, the activation layer, and the batch normalization (BN) layer. While pooling layers are commonly employed in CNN to reduce network parameters and computational complexity, it has been omitted in the RDN module to prevent feature map shrinkage and the loss of noise feature information during the training process. Moreover, removing the pooling layer ensures a constant feature map size since only feature maps of the same size can be computed at the output of the residual learning module.

In order to transform the complex-valued input into the real-valued input, we separate the real and imaginary parts of $\boldsymbol{Y}$ and stack them together, resulting in the final input array $\boldsymbol{Y}_1 \in \mathbb{R}^{2\times r\times q}$. To extract time-frequency characteristics of received signal, different layers have been combined into three blocks.

The first layer of Block1 performs the convolution operation on $\boldsymbol{Y}_1$, which can be expressed as

$$\boldsymbol{Z} = Conv(\boldsymbol{Y}_1) = \boldsymbol{Y}_1 * \boldsymbol{K} + \boldsymbol{b}_1, \tag{3}$$

where $\boldsymbol{K}$ represents the weights of the convolution kernel and $\boldsymbol{b}_1$ denotes the bias with $*$ being the convolutional operator. Then, the rectified linear unit (ReLU) is designed for nonlinear transformation and can be represented as

$$ReLU(\boldsymbol{Z}) = \max(0, \boldsymbol{Z}), \tag{4}$$

Block2 consists of the Conv layer, the ReLU layer and the BN layer. The BN layer has been inserted between the Conv layer and the ReLU layer to expedite the training process:

$$BN(\boldsymbol{Z}_i) = \boldsymbol{\gamma}\frac{\boldsymbol{Z}_i - \boldsymbol{\mu}_B}{\sqrt{\boldsymbol{\sigma}_B^2 + \varepsilon}} + \boldsymbol{\beta}, \tag{5}$$

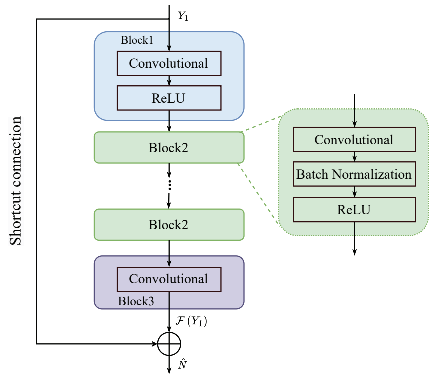

Fig. 2. Network architecture of the proposed RDN module.

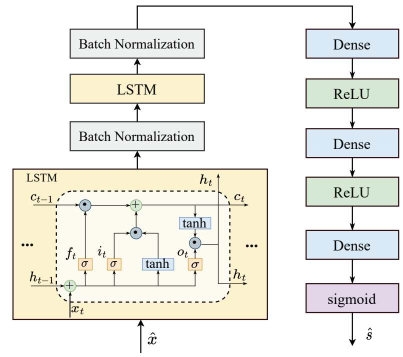

Fig. 3. Network architecture of the proposed SLSTM module.

where $\boldsymbol{Z}_i$ is the input of the ith BN layer, and $\boldsymbol{\mu}_B$ and $\boldsymbol{\sigma}_B^2$ denote the mean and variance calculated from the mini-batch samples, respectively. Here, $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are the continuously updated scale and bias factors during training, and ε is a very small value to avoid the denominator being zero.

Finally, Block3 obtains the feature map $\mathcal{F}(\boldsymbol{Y}_1)$ through the Conv layer and reconstructs the output by merging $\mathcal{F}(\boldsymbol{Y}_1)$ with the input via a shortcut connection. The output of the RDN module is expressed as

$$\hat{\boldsymbol{N}} = \mathcal{H}(\boldsymbol{Y}_1) = \boldsymbol{Y}_1 + \mathcal{F}(\boldsymbol{Y}_1). \tag{6}$$

C. Network Architecture of SLSTM Module

Traditional LSTM network comprises three layers: the input layer, the LSTM layer (hidden layer) and the output layer. The fundamentals component of the LSTM layer is the memory block, which contains multiple cyclically connected cells. Each cell consists of an input gate $\boldsymbol{i}_t$, a forget gate $\boldsymbol{f}_t$, and an output gate $\boldsymbol{o}_t$. Specifically, $\boldsymbol{i}_t$ is responsible for learning how to store information in memory, $\boldsymbol{f}_t$ controls the duration of the stored information, and $\boldsymbol{o}_t$ determines when to utilize the stored information. The hidden state $\boldsymbol{h}_t$ of a single-layer LSTM is computed recursively using the following equations:

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma\left(\boldsymbol{W}_{xi}\boldsymbol{x}_t + \boldsymbol{W}_{hi}\boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right),\\
\boldsymbol{f}_t &= \sigma\left(\boldsymbol{W}_{x\mathrm{f}}\boldsymbol{x}_t + \boldsymbol{W}_{hf}\boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right),\\
\boldsymbol{c}_t &= \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ tanh(\boldsymbol{W}_{xc}\boldsymbol{x}_t + \boldsymbol{W}_{hc}\boldsymbol{h}_{t-1} + \boldsymbol{b}_c),\\
\boldsymbol{o}_t &= \sigma\left(\boldsymbol{W}_{xo}\boldsymbol{x}_t + \boldsymbol{W}_{ho}\boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right),\\
\boldsymbol{h}_t &= \boldsymbol{o}_t \circ tanh(\boldsymbol{c}_t),
\end{aligned} \tag{7}$$

where $\boldsymbol{x}_t$ represents the input vector and $\boldsymbol{c}_t$ denotes the cell state. σ and $tanh$ are nonlinear activation functions defined as $\frac{1}{1+e^x}$ and $\frac{e^x-e^{-x}}{e^x+e^{-x}}$, respectively. The symbol $\circ$ denotes the Hadamard product, and $\boldsymbol{W}$ and $\boldsymbol{b}$ indicate the learnable network parameters, respectively.

To efficiently extract characteristics of the transmitted signal, the SLSTM network is employed to increase the depth of network and handle more complex problems. The mapping function from the input of the LSTM layer to the hidden state is denoted as $\mathcal{G}$, and the hidden state $\boldsymbol{h}_t^l$ of the lth LSTM layer is obtained by recursively iterating the following equation

$$\boldsymbol{h}_t^l = \mathcal{G}\left(\boldsymbol{W}_t^l \boldsymbol{h}_t^{l-1} + \boldsymbol{W}_{t-1}^l \boldsymbol{h}_{t-1}^l + \boldsymbol{b}^l\right), \tag{8}$$

where $\boldsymbol{h}_{\mathrm{t}}^0 = \boldsymbol{x}_t$.

The SLSTM module employs a two-layer stacked LSTM architecture, as depicted in Fig. 3. The first LSTM layer produces a sequence output that serves as the input to the subsequent LSTM layer. The BN layers are incorporated after each LSTM layer to expedite network convergence and mitigate gradient vanishing. The final output vector $\boldsymbol{h}_t^L$ contains comprehensive information of the input vectors, which are then passed to the subsequent layer for training data fitting.

The Dense layer consolidates the extracted local features to derive the global features and enhance the robustness of the network, which can be expressed as

$$Dense(\boldsymbol{h}_t^L) = \boldsymbol{W}\boldsymbol{h}_t^L + \boldsymbol{b}_2, \tag{9}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}_2$ denote the weight and bias of the Dense layer, respectively. To mitigate the problem of overfitting and reduce the interdependence between parameters, the ReLU activation function is further employed after the Dense layer.

The output layer is configured with m output nodes employing the sigmoid activation function to obtain an output vector $\boldsymbol{p} = [p_1, ..., p_m]^T$, where each element p_i represents the probability of ith bit $\hat{s}_i$ being recovered as 1. By applying a probability threshold of α, the predicted bit stream $\hat{\boldsymbol{s}}$ is obtained through a decision process based on the vector $\boldsymbol{p}$, which is defined as

$$\hat{s}_m = \begin{cases} 1, & p_i \geq \alpha \\ 0, & p_i < \alpha \end{cases}. \tag{10}$$

IV. Simulation Results

A. Dataset Generation

In the simulation, a 32-bits stream is randomly generated, encoded with (7,4) Hamming codes, modulated by quadrature phase shift keying (QPSK), and then pulse-shaped using a root-raised cosine filter with a roll-off factor of 0.5. The received signal is sampled at a rate 8 times the symbol rate and the STFT was performed using a Hamming window with the window length of 3 sampling points. The probability threshold α is set to 0.5. The effects of both AWGN and IQ imbalance are discussed. Here, IQ imbalance is quantified by the parameters (α, β), where α represents the amplitude imbalance in dB and β denotes the phase imbalance in degrees. Two IQ imbalance configurations, (-3,-2) and (5,10), are considered, respectively.

Different training and testing data are used for the RDN and SLSTM modules as shown in Table I. The training E_b/N_0

TABLE I
SETTINGS FOR DATA GENERATION

	RDN module	SLSTM module
Training E_b/N_0 range	0:1:8dB	0:1:8dB
Training samples per E_b/N_0	200000	200000
Testing E_b/N_0 range	0:1:8dB	0:0.5:8dB
Testing samples per E_b/N_0	100000	100000

TABLE II
NETWORK PARAMETER OF THE RDN AND SLSTM MODEL

Module	Layer(types)	Hyperarameters
RDN module	Conv(Block1)	Kernel = 64, Kernel size = (3,3)
	Conv(Block2)	Kernel = 64, Kernel size = (3,3)
	Conv(Block3)	Kernel = 64, Kernel size = (3,3)
SLSTM module	LSTM	Neurons=32
	LSTM	Neurons=32
	Dense	Neurons=256
	Dense	Neurons=128
	Dense	Neurons=32

ranges from 0 dB to 8 dB, with an interval of 1 dB. Moreover, to test the adaptability of the scheme to the untrained E_b/N_0, the testing data for the SLSTM module covered E_b/N_0 values from 0 dB to 8 dB, with a 0.5 dB interval. The testing set of the RDN module corresponds to that of the SLSTM module with a 1 dB interval and consists of corresponding STFT results to verify the denoising performance.

B. Training Model

The network parameters for the proposed RDN and SLSTM modules are presented in Table II. The hyperparameters of the proposed model are determined based on a commonly used grid search method. Specifically, the grid search method involves defining a set of possible values for each hyperparameter that requires tuning. Then, the proposed model is trained with each possible combination of hyperparameters. Finally, the optimal hyperparameters that achieve the best BER performance are selected for the proposed model. To obtain sufficient time-frequency characteristics for the denoising processing, the repetition number of Block2 is set to 25. The network is trained by the Adam method with a batch size $N_B = 512$ and an initial learning rate of 0.001. The loss function measures the disparity between the output and the correct label, and the network parameters are optimized by minimizing the loss function. To prevent overfitting, the learning rate is dynamically adjusted to $1/2$ of the original value when the validation set loss plateaus. Additionally, the epoch size is adaptively determined to achieve better generalization performance.

For the RDN module, the objective is to minimize the discrepancy between $\boldsymbol{N}$ and $\hat{\boldsymbol{N}}$, which can be accomplished by employing the mean square error (MSE) as the loss function:

$$L_1 = \frac{1}{2N_B} \sum_{i=1}^{N_B} \left\| \hat{\boldsymbol{N}}_i - \boldsymbol{N}_i \right\|^2, \tag{11}$$

where $\|\cdot\|$ denotes the Euclidean norm.

The SLSTM module aims to learn a mapping function that can recover the original information. This process can be seen

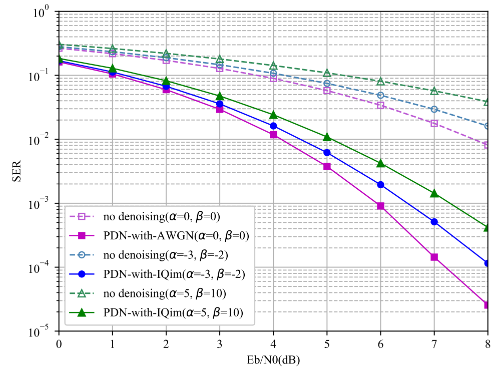

Fig. 4. The SER performance of the proposed PDN module under hard decision, with AWGN and IQ imbalance.

as a binary classification problem, where each node in the output layer produces a binary output of 0 or 1. The commonly used loss function in this case is the cross entropy, given by

$$L_2 = -\frac{1}{N_B m} \sum_{i=1}^{N_B} \sum_{j=1}^{m} \left[s_{ij} \log\left(p_{ij}\right) + \left(1 - s_{ij}\right) \log(1 - p_{ij}) \right], \tag{12}$$

where p_{ij} denotes the predicted value of the jth neuron in output layer, representing the probability that the jth bit of $\hat{\boldsymbol{s}}$ is predicted as 1. Here, the true label corresponding to the jth bit is denoted by s_{ij}.

C. Denoising Capacity Comparison

To evaluate the effectiveness of the RDN module, we conduct a comparison of Peak Signal-to-Noise Ratio (PSNR) before and after denoising under various E_b/N_0 scenarios. Additionally, the impact of the IQ imbalance on the received signal after performing STFT is also considered. The PSNR results are presented in Table III, indicating significant denoising capabilities of the RDN module, particularly in the moderate and high E_b/N_0 conditions.

The symbol error rate (SER) performance of the proposed PDN module has been verified in Fig. 4. We also include the result of no denoising scheme by directly performing hard decision using $\boldsymbol{y}$ instead of $\hat{\boldsymbol{x}}$ for comparison. It can be observed that: 1) the proposed RDN module can remarkably improve the SER performance and a performance gap of around 3.2 dB between no denoising scheme and our scheme can be observed under moderate and high E_b/N_0 conditions. 2) When the IQ imbalance is considered, the performance gain of PDN-with-IQim(-3, -2) is 3.5 dB, while PDN-with-IQim(5, 10) yields a performance gain of approximately 4.7 dB.

D. BER Performance Comparison

We compare the BER performance of PDN-SLSTM scheme with two baseline approaches. The first baseline employs the Hamming decoder with a hard decision, which is referred to as the ideal hard decision scheme. The second one is the ideal maximum likelihood (ML) decision scheme, which performs ML decoding when only AWGN is present. Both baseline results are derived from extensive data-driven simulations to obtain their respective error rates.

TABLE III
THE AVERAGE PSNR RESULTS CORRESPONDING TO DIFFERENT E_b/N_0 UNDER NON-IDEAL CONDITIONS

		0dB	1dB	2dB	3dB	4dB	5dB	6dB	7dB	8dB
AWGN	no denoising	4.80	5.22	5.68	6.17	6.68	7.23	7.81	8.42	9.06
	after RDN module	13.13	16.35	19.15	24.82	31.93	39.11	43.64	45.93	46.91
IQ imbalance(-3,-2)	no denoising	4.73	5.15	5.60	6.08	6.59	7.13	7.70	8.29	8.92
	after RDN module	12.86	14.82	18.08	22.02	29.34	36.37	41.53	44.42	45.79
IQ imbalance(5,10)	no denoising	4.60	5.00	5.43	5.89	6.38	6.89	7.42	7.98	8.56
	after RDN module	12.48	14.01	17.10	19.41	25.29	31.07	35.56	37.55	39.01

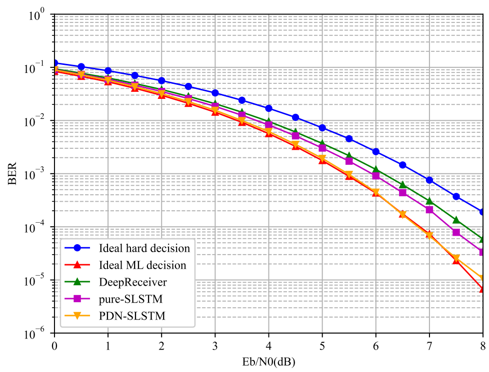

Fig. 5. The BER performance of the PDN-SLSTM, pure-SLSTM compared to the DeepReceiver and baselines.

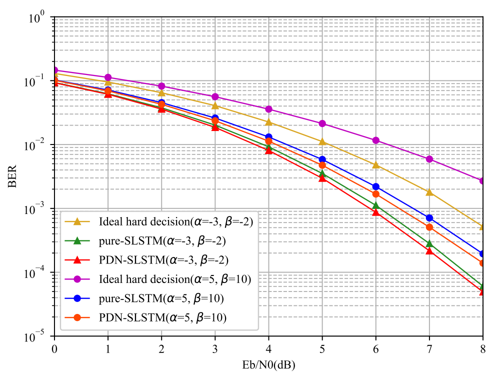

Fig. 6. The BER performance of the PDN-SLSTM, pure-SLSTM and the hard decision scheme.

Fig. 5 illustrates the performance of the PDN-SLSTM, pure-SLSTM, DeepReceiver [12] and the baselines. The pure-SLSTM represents the case without the PDN module for denoising. Notably, the pure-SLSTM achieves significantly lower BER compared to the ideal hard decision scheme and DeepReceiver. The PDN-SLSTM scheme further improves the BER performance and approximates the ideal ML decision scheme, approaching the optimal performance. Moreover, both pure-SLSTM and PDN-SLSTM schemes exhibit stable BER performance on untrained E_b/N_0 values, indicating their strong generalization ability.

Fig. 6 evaluate the performance of the pure-SLSTM and PDN-SLSTM schemes when the IQ imbalance is considered. It is evident that the performance of the ideal hard decision scheme decreases significantly with the increment of IQ imbalance, and the pure-SLSTM is less affected due to its capability of IQ imbalance correction. Additionally, the PDN-SLSTM scheme outperforms the pure-SLSTM in terms of BER performance. The results confirm the advantages of the PDN-SLSTM scheme, even in the presence of IQ imbalance.

V. CONCLUSION

In this correspondence, we developed a novel intelligent PL receiver based on PDN-SLSTM scheme to improve the reliability of wireless communication systems. The PDN module was first proposed for efficient denoising based on the CNN and residual learning. Then, the SLSTM module was carefully designed to achieve accurate information recovery. The effectiveness of the PDN-SLSTM scheme has been confirmed through simulation results. It not only outperforms the baselines but also demonstrates robustness to the IQ imbalance.

REFERENCES

[1] S. He, K. Shi, C. Liu, B. Guo, J. Chen, and Z. Shi, "Collaborative sensing in internet of things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 3, pp. 1435–1474, Jun. 2022.

[2] Z. Sha and Z. Wang, "Channel estimation and equalization for terahertz receiver with RF impairments," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 6, pp. 1621–1635, Apr. 2021.

[3] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, Apr. 2019.

[4] J. Liu, H. Zhao, D. Ma, K. Mei, and J. Wei, "Theoretical analysis of deep neural networks in physical layer communication," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6589–6603, Aug. 2022.

[5] M. Honkala, D. Korpi, and J. M. J. Huttunen, "Deeprx: Fully convolutional deep learning receiver," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3925–3940, Jun. 2021.

[6] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May. 2015.

[7] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Feb. 2019.

[8] Q. Bai, J. Wang, Y. Zhang, and J. Song, "Deep learning-based channel estimation algorithm over time selective fading channels," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 125–134, Sep. 2020.

[9] A. Caciularu and D. Burshtein, "Unsupervised linear and nonlinear channel equalization and decoding using variational autoencoders," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 6, no. 3, pp. 1003–1018, Apr. 2020.

[10] L. Zhang, H. Zhang, Y. Jiang, and Z. Wu, "Intelligent and reliable deep learning LSTM neural networks-based ofdm-dcsk demodulation design," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16 163–16 167, Sep. 2020.

[11] J. Dai, K. Tan, Z. Si, K. Niu, M. Chen, H. V. Poor, and S. Cui, "Learning to decode protograph LDPC codes," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1983–1999, May. 2021.

[12] S. Zheng, S. Chen, and X. Yang, "Deepreceiver: A deep learning-based intelligent receiver for wireless communications in the physical layer," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 5–20, Aug. 2021.

[13] W.-H. Lee, M. Ozger, U. Challita, and K. W. Sung, "Noise learning-based denoising autoencoder," *IEEE Commun. Lett.*, vol. 25, no. 9, pp. 2983–2987, Jun. 2021.

[14] X. Yang, L. Zhang, Y. Feng, and Z. Wu, "A residual cnn-based denoiser for reliable recovery of bit stream with applications to soft channel decoding," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 8, no. 4, pp. 1677–1688, Aug. 2022.