Autoencoder based Robust Transceivers for Fading Channels using Deep Neural Networks

Sandesh Rao Mattu[1], Lakshmi Narasimhan T[2], and A. Chockalingam[1]

[1]Department of ECE, Indian Institute of Science, Bangalore 560012

[2]Department of EE, Indian Institute of Technology, Palakkad 678623

***Abstract*—In this paper, we design transceivers for fading channels using autoencoders and deep neural networks (DNN). Specifically, we consider the problem of finding (n, k) block codes such that the codewords are maximally separated in terms of their Hamming distance using autoencoders. We design an encoder and robust decoder for these block codes using DNNs. Towards this, we propose a novel training methodology for the DNN that attempts to maximize the minimum Hamming distance between codewords. We propose a loss function for this training which has stable weight updates during back propagation compared to other loss functions reported in the literature. The block codes learned using the proposed methodology are found to achieve the maximal Hamming distance separation that is known in theory. We also propose two different receiver architectures based on fully connected deep neural network (FCDNN) and bi-directional recurrent neural network (BRNN) that are suited for complex fading channels. The proposed DNN based receiver is shown to achieve significantly better error performance when compared to their classical counterparts in the presence of channel model mismatches. In the presence of model mismatches such as imperfect channel knowledge and noise correlation, the proposed DNN based transceiver is shown to offer increased reliability and robustness than the conventional transceiver.**

Keywords – *Deep neural networks, linear block codes, autoencoder, correlated noise, Gaussian noise, fading, transceiver.*

I. Introduction

Recently, deep neural networks (DNN) have shown promising performance in inference tasks in several fields [1]. A DNN needs to be trained before it can be employed for inference tasks. A trained DNN requires relatively less computational complexity for performing the inference tasks compared to conventional optimal statistical inference methods. DNNs have been used to build efficient wireless communication systems [2]. With the advent of sophisticated software tools and optimized hardware for machine learning, the computational complexity of machine learning algorithms have become practical. Current generation mobile phones and computational devices are built with fast general purpose neural networks in the hardware, which can be configured to perform signal processing tasks for communication in real-time. In this context, we study the design of a transmitter and receiver using DNN whose performance is robust to channel model mismatches. Two primary model mismatches that often occur in practice are imperfect channel state information (CSI) and colored noise. In the design of the transceiver, we focus on improving the reliability of the system under practical conditions. To this end, we use DNNs to design: (*i*) block codes over binary field for forward error correction (FEC), and (*ii*) receivers that are robust to imperfect CSI and noise correlation.

The metric used in the literature to quantify the performance of block codes is the Hamming distance separation between the codewords [3]. For given dimensions, the block codes which maximize the Hamming distance separation between their codewords achieve the best performance. The minimum distance decoder (MDD) [3] based on the Euclidean distance is the conventionally used decoder for block codes. It can also be shown that MDD is the optimal maximum likelihood receiver when the noise in the communication channel is additive white Gaussian noise (AWGN). A disadvantage of MDD is that its computational complexity is exponential in the dimension of the input message.

In the literature, neural networks have been employed to build decoders for block codes [4]. In [2], [5], autoencoders were employed to design FEC codes and corresponding encoders and decoders. The design of machine learning based transceivers for AWGN channels was studied in [2]. The use of convolutional neural networks (CNN) and recurrent neural networks (RNN) for digital demodulation was studied in [6]. Demodulation of signals through a fading channel using neural networks was reported in [7], [8]. Further, the design of constellations for communication in AWGN channels using DNNs was studied in [2], [9]. Deep learning based demodulation for MIMO fading channels has been reported in [10]. However, to the best of our knowledge, the design of robust transceivers with block codes using DNNs for fading channels has not been reported in the literature so far. Our new contributions in this paper can be summarized as follows.

- We propose an autoencoder based DNN to design block codes, encoder, and decoder.
- We propose a novel training methodology for the DNN to obtain block codes with maximal Hamming distance separation that meets the theoretical upper bound. We propose a loss function for this training.
- We propose two receiver architectures based on fully connected deep neural network (FCDNN) and bi-directional recurrent neural network (BRNN) for fading channels.
- Finally, we show that the proposed DNN based transceiver is robust and it outperforms the conventional transceiver in the presence of channel model mismatches.

II. Problem Formulation

Consider a point-to-point wireless communication system with a single-antenna transmitter and receiver. At the transmitter, the input message $\mathbf{m}$ of k bits is encoded using a block code to a codeword $\mathbf{c}$ of length n bits ($n > k$). The encoded bits are modulated using a constellation $\mathbb{A}$ (e.g., QAM, PSK) to obtain the transmit signal $\mathbf{x}$. The symbols in $\mathbf{x}$ are transmitted serially over the wireless channel. Let h be the fading channel gain of the wireless channel. The received signal is given by $\mathbf{y} = h\mathbf{x}\ \mathbf{n}$, where $\mathbf{n}$ is the noise vector and h is assumed to be constant over the transmission period of a codeword.

This work was supported in part by the J. C. Bose National Fellowship, Department of Science and Technology, Government of India, and the Intel India Faculty Excellence Program.

Conventionally, the noise is modeled as independent and identically distributed zero mean complex Gaussian random variable with some variance denoted by σ^2. However, in practice, the noise in devices can become colored or correlated [11]. In [12], the authors show that the auto-correlation of such colored noise, referred to as flicker noise, is given by

$$R_{nn}(t,\tau) = \begin{cases} \frac{A}{(1-t/T_o)^{\frac{1}{\alpha}}} \cos \frac{c\tau}{(1-t/T_o)^{\frac{1}{\alpha}}} & \text{if } \begin{matrix} t \in (0, T_o) \ \& \\ \tau \in (-T_o, T_o) \end{matrix} \\ 0 & \text{elsewhere} \end{cases}, \quad (1)$$

where T_o is the period of observation, c is the lowest fluctuation frequency, α and A are positive constants that depend on the hardware device characteristics. For practical values of these parameters, it can be seen that the correlation of noise samples over the period of a codeword remains almost constant [12]. Thus, the covariance matrix of the correlated noise $\mathbf{n}$ can be given by $(1-\rho)\mathbf{I}\,\rho\mathbf{1}$, where ρ is the correlation coefficient computed from (1), $\mathbf{I}$ is an identity matrix and $\mathbf{1}$ is a matrix of all ones. For $\rho = 0$, we get the case of the i.i.d. AWGN.

We assume a quasi-static flat fading channel. The fade coefficients h are modeled to be a complex circularly symmetric normal random variable $\mathcal{CN}(0,1)$. To recover $\mathbf{x}$ from $\mathbf{y}$, the conventional receiver computes

$$\frac{h^*\mathbf{y}}{|h|^2} = \mathbf{x} \ \ \frac{h^*\mathbf{n}}{|h|^2}, \quad (2)$$

where h^* is the complex conjugate of h. The receiver employs MDD to obtain the transmitted message from (2). Hence, the knowledge of channel gain (CSI) is required for decoding at the receiver. However, in practice, perfect CSI may not be available. The channel gains are estimated before the decoding operation using pilot information. The error in CSI can be modeled using the distribution of the noise at the receiver.

In a practical receiver, the conventional assumption of AWGN and perfect CSI may not be true. Such assumptions can adversely affect the system performance. Therefore, in this paper, we design block codes, encoders, and decoders such that they are robust to the channel model mismatches described above. In the following, we present the proposed autoencoders based solution to this problem using DNNs.

III. Proposed DNN Architecture and Training Methodology

In this section, we present a technique to design block codes using autoencoders that achieve near-optimal performance. We also present a DNN based robust receiver for fading channels.

A. Code design using autoencoders

Here, we design a binary code such that each k bit long message is mapped on to an n bit long codeword. The Hamming distance between two codewords $\mathbf{c}_i$ and $\mathbf{c}_j$ is

$$d_H(\mathbf{c}_i, \mathbf{c}_j) = \left|\{k : c_{i,k} \neq c_{j,k}\}\right|, \quad (3)$$

where $|\cdot|$ denotes cardinality, and $c_{i,k}$ and $c_{j,k}$ are the kth elements in codewords $\mathbf{c}_i$, $\mathbf{c}_j$, respectively. The minimum Hamming distance of a code C is computed as

$$d_{\min}(C) = \min_{\substack{\mathbf{c}_i, \mathbf{c}_j \in C \\ \mathbf{c}_i \neq \mathbf{c}_j}} d_H(\mathbf{c}_i, \mathbf{c}_j). \quad (4)$$

The number of errors that can be corrected by a code is given by $t = \lfloor \frac{d_{\min}-1}{2} \rfloor$. Hence, it is desirable to maximize $d_{\min}(C)$ to improve the performance of the code. We shall employ autoencoders to design block codes with maximum $d_{\min}$.

Autoencoders consist of two neural networks connected back to back, and they are traditionally used to obtain lower dimensional representation of the input data. In traditional autoencoders, the first neural network represents or encodes the input data into a lower dimensional output and the second neural network decodes this compressed data to recover the original data [13]. In the proposed setup, the first neural network encodes or represents the input signal with a high dimensional output, subsequently channel distortions are introduced, and the second neural network maps this high dimensional signal after channel distortions to the input signal space. The input of the autoencoder is the k length message. The output of the autoencoder are 2^k length one-hot vectors [14] corresponding to the input vectors. The encoder and decoder networks are a sequence of fully connected layers.

The autoencoder is trained in the following manner. The 2^k possible messages are input to the encoder neural network in a random order. The encoder neural network layers are built with the tanh activation function (c.f. (5)). The output from the encoder neural network is passed through the wireless channel. The output of the wireless channel is fed to the decoder neural network. We use the softmax activation function $(\bar{\sigma}(\mathbf{z})_i$, where $i = 1, \ldots, n)$ at the final layer of the decoder network. The activation functions are

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) \ \ \exp(-x)}; \quad \bar{\sigma}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_{j=1}^{n} \exp(z_j)}, \mathbf{z} \in \mathbb{R}^n. \quad (5)$$

This training is performed for several instances of $\mathcal{CN}(0,1)$ r.v. (fading case), and training is carried out with decreasing values of signal-to-noise ratio (SNR). The n-length bit sequence corresponding to the output 2^k one-hot vectors are the learned codewords.

We propose to train the autoencoder by optimizing its weights in two stages. In order to maximize the Hamming distance separation of the codewords, the first stage optimization for the autoencoder is performed only for updating the weights of the encoder network. We use the following loss function for the first stage optimization: $L_e(\theta_e) \triangleq -\lambda * d_{\min}(\theta_e)$, where θ_e are the weights of the encoder neural network and λ is the regularization parameter. The negative sign in the loss function ensures that the minimization of this loss leads to the maximization of $d_{\min}$. The second stage optimization updates the weights of both the encoder and decoder networks using the binary cross entropy as the loss function. The binary cross entropy loss function is given by $H(\mathbf{p}, \mathbf{q}, \Theta) = -\mathbb{E}_{\mathbf{p}}\left[\log \mathbf{q}(\Theta)\right]$, where $\mathbf{p}$ is the expected probability mass function (PMF), $\mathbf{q}$ is the output PMF , and Θ is the weight matrix. Although the second stage optimization updates the weights at the encoder that are obtained from the first stage, the gradients computed in the second stage are dependent on the output of the first stage which ensures maximal codeword separation.

In the first stage, conventional method of computing $d_{\min}$ through pairwise combinations results in loops and stability issues in the back propagation algorithm used to optimize the weights of the neural network. We propose the following efficient and stable method to compute the $d_{\min}$.

Let $\mathbf{C}$ be the matrix whose columns are the codewords. Using the squared pairwise Euclidean distance between the codewords, we define a distance matrix $\mathbf{D}$. The (i, j)th element of $\mathbf{D}$ is defined as

$$\mathbf{D}(i, j) = (\mathbf{c}_i - \mathbf{c}_j)(\mathbf{c}_i - \mathbf{c}_j)^T = \mathbf{c}_i\mathbf{c}_i^T - 2\mathbf{c}_i\mathbf{c}_j^T \;\; \mathbf{c}_j\mathbf{c}_j^T, \qquad (6)$$

where $\mathbf{c}_i$ is a row vector and the ith codeword. Let $\mathbf{L}$ be a matrix whose (i, j)th element is the squared norm of the ith codeword. Now, (6) can be simplified to get $\mathbf{D} = \mathbf{L} - 2\mathbf{C}\mathbf{C}^T$ $\mathbf{L}^T$. Since the diagonal entries in the matrix $\mathbf{D}$ correspond to $i = j$, diagonal elements are zero. The $d_{\min}$ is proportional to the minimum off-diagonal element of $\mathbf{D}$. The $d_{\min}$ is computed through computation of this matrix. This formulation also ensures that the computations are differentiable either through direct gradients or pseudo-gradients.

Note that though the distance properties of the code were used in the loss function in [5], it was employed for training the entire network including both the encoder and decoder. However, in our proposed setup, the loss function $L_e(\theta_e)$ is employed to train only the encoder. Further, the minimum distance of only a subset of the codewords is computed in [5]. Whereas, we compute the pairwise distances between all codewords. This helps us to design block codes with better Hamming distance properties than the method in [5].

Figure 1 illustrates the neural network described above. In Fig. 1, m_is are the input message bits, c_is form the output codeword, c_i's are output of the channel, and $\hat{m}_i$s are the decoded one-hot vectors.

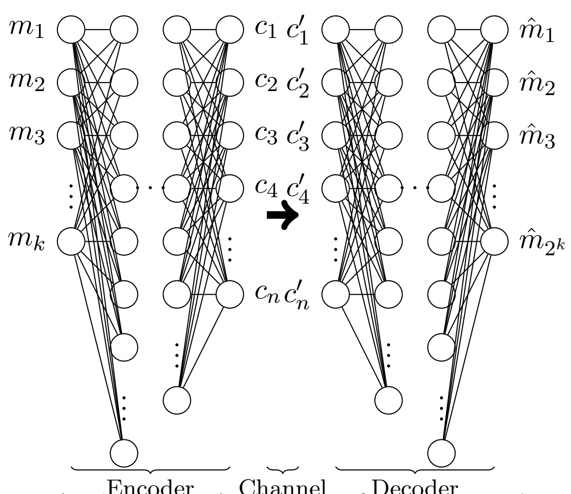


Fig. 1: Schematic diagram of the autoencoder.

B. Design of robust receivers for fading channels

Traditional neural networks operate on data from the real field. However, in our case, we need to design neural network architectures for complex fading channels. For this purpose, we propose the following two architectures.

1) Fully connected deep neural networks based receiver: In this architecture, we employ FCDNN with a pre-processing layer that performs the operation in (2) on the received data with the knowledge of CSI. The real and imaginary components of the pre-processed data are stacked together and fed as input to the network. The output of the decoder network is a one-hot vector. This network is trained by minimizing the binary cross-entropy loss for different realizations of the flat-fading complex channel at decreasing SNRs. A block diagram of this FCDNN model is shown in Fig. 2, where c' denotes the symbols received from the channel and h denotes the complex channel gains. The advantage of the model is its simplicity and low complexity. However, due to pre-processing, the neural network may not learn the correlation that could be present between the received data and CSI. We propose a bidirectional recurrent neural network (BRNN) architecture to alleviate this disadvantage.

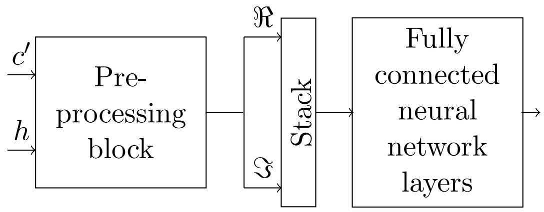


Fig. 2: Block diagram of the FCDNN decoder.

2) Bidirectional recurrent neural networks based receiver: The recurrent neural network (RNN) [15] is a class of neural networks which has a temporal growth in one dimension. These neural networks unfold in time to capture any time-dependent patterns. The principle of BRNN is to split the neurons of a conventional RNN into two directions, one for forward direction and another for backward direction. Input to the BRNN decoder is the output from the channel. The input to the BRNN is a multi-dimensional array. The real and imaginary components of the received signal are stacked along with the real and imaginary components of the channel gains in the input multi-dimensional array in different dimension. The output of the BRNN decoder is a one hot vector. The BRNN is trained by minimizing the binary cross-entropy loss for different realizations of the flat-fading complex channel at decreasing SNRs. When perfect CSI is available at the receiver, the exact channel gains are fed as input to the BRNN. However, in practical scenarios, only an estimated CSI would be available at the receiver, which is fed as the input to the BRNN. Therefore, in the training process, the BRNN is trained using both perfect and imperfect CSI data to build a robust receiver. The advantage of BRNN over FCDNN is the reduced computational complexity due to the absence of pre-processing. Further, any time correlation in the channel can learned by the BRNN, thereby, providing robust performance.

The performance of the proposed autoencoder based FCDNN and BRNN transceivers in AWGN and fading channels are presented in the next section.

IV. Results and Discussions

In this section, we present the block error rate (BLER) performance of the proposed DNN based receivers. The BLER at the receiver is defined as the ratio of the number of FEC blocks in error to the total number of FEC blocks transmitted. We show the performance of the block codes designed by the proposed DNN using autoencoders for different block sizes. The configuration of the DNN used to build the encoders for different code sizes (n, k) is listed in Table I. The activation function used at all the layers of the encoder is tanh. The configurations of the FCDNN and BRNN decoders are listed in Tables II and III, respectively. The final layer of the decoder neural network produces the one-hot vectors (of size 2^k) using the softmax activation function as described in Sec. III-A. The rest of the layers in the decoder use the tanh activation function. These neural networks are trained as described in Sec. III with batches of size 2^k for an (n, k) code. The encoder produces an n-length codeword for each k-length input. These codewords are passed through the channel and its output with additive noise is given to decoder for decoding into one-hot vectors. This training is performed for different channel realizations and SNR values.

A. Performance of the proposed receiver in AWGN channel

The performance of the block codes designed by the proposed DNN architecture is shown in Fig. 3. We compare the performance of the block codes learned by the proposed DNN with that of the optimal block codes for a given block size from [3]. It can be see that the codes learned by the proposed DNN perform almost as good as the theoretical optimum block codes with MDD in the AWGN channel. From Table IV, we can also see that the proposed loss function based training achieves the best d_{min} value for the codes learned by the neural network.

TABLE I: Architecture of the encoder neural networks

Code size (n, k)	No. of layers	Input dimension of each layer	Output dimension of each layer
(7,4)	3	4, 32, 16	32, 16, 7
(15,11)	4	11, 90, 45, 30	90, 45, 30, 15
(21,11)	5	11, 300, 200, 100, 50	300, 200, 100, 50, 21

TABLE II: Architecture of the FCDNN based decoder

Code size (n, k)	No. of layers	Input dimension of each layer	Output dimension of each layer
(7,4)	2	7, 16	16, 16
(15,11)	2	15, 2048	2048, 2048
(21,11)	2	21, 2048	2048, 2048

TABLE III: Architecture of BRNN based decoder

Hidden units	100
Time steps	7
Output dimension	2^k

TABLE IV: Distance properties of the learned codes

Code size (n, k)	Theoretical maximum d_{min}	d_{min} achieved by the learned code
(7,4)	3	3
(15,11)	3	3
(21,11)	6	6

Robustness: To analyze the robustness of the proposed transceiver to mismatches in channel model, we evaluate their BLER performance in the presence of correlated noise. The neural networks are trained with correlated noise at different SNRs and correlation coefficients. We compare the performance of the proposed autoencoder based transceiver with that of the conventional transceiver with MDD at different levels of model mismatches, i.e., for different values of the noise correlation coefficient ρ. Figures 4a and 4b show that the performance of the proposed autoencoder based transceiver

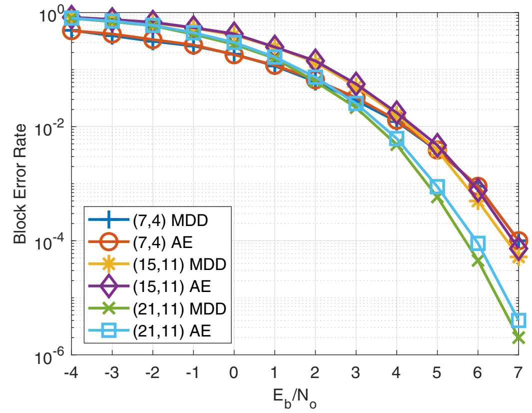


Fig. 3: BLER performance comparison between the proposed autoencoder (AE) based block codes with FCDNN and optimal block codes with MDD in AWGN channel, and $h = 1$.

is superior than that of the conventional transceivers at correlation coefficients of 0.3 and 0.5, and different block sizes. For example, at an SNR of 4 dB, block size of (21, 11), the neural network based receiver achieves a BLER of about 10^{-3} when the correlation coefficient is 0.3 while the conventional receiver achieves only a BLER of about 0.03. Interestingly, the BLER of the neural network based receiver decreases with the increase in the noise correlation, while that of the conventional receiver increases with the noise correlation. Note that the neural network based receiver performs the decoding without any knowledge of the noise correlation. Thus, it can be seen that the proposed autoencoder based transceivers using DNN are quite robust to model mismatches such as noise correlation.

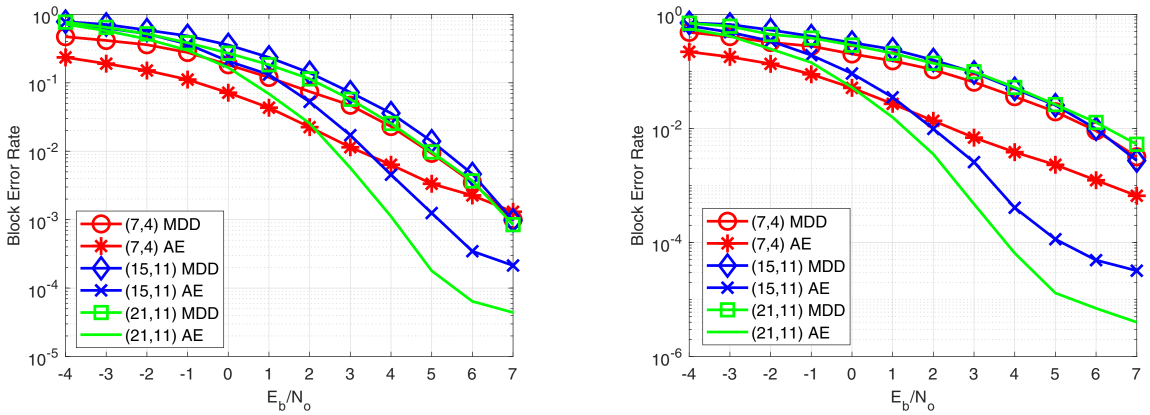
B. Performance of the proposed receiver in fading channel

In Fig. 5, we compare the performance of the proposed FCDNN and BRNN based transceivers with that of the conventional transceiver with the pre-processing in (2) and MDD for two block codes in Rayleigh fading channel with perfect CSI. It can be seen that both the FCDNN and BRNN based receivers perform similarly. At low SNRs, the FCDNN based receiver and the conventional receiver have similar performance. At high SNRs, the conventional receiver performs slightly better than the neural network based receivers.

Robustness: To analyze the robustness of the proposed receivers to model mismatches, we evaluate their BLER performance in the presence of imperfect CSI and correlated noise. The model mismatch due to imperfect CSI is measured by the mean square error (MSE) in the estimate of the channel gain. In Fig. 6, we observe that the performance of the proposed DNN based receivers is superior than that of the conventional receiver for different levels of model mismatches (i.e., MSE of channel estimate and noise correlation coefficient) and a block size of (7,4). Further, at high SNRs, the BRNN based receiver achieves better BLER than the rest of the receivers. As before, the receivers are unaware of the mismatches in the channel model. Thus, we see that the neural network based receivers provide better reliability than the conventional receivers and are robust to practical model mismatches.

V. Conclusions

Conventional transceivers for AWGN and fading channel provide optimal performance when the channel models are



(a) $\rho = 0.3$ (b) $\rho = 0.5$

Fig. 4: BLER performance comparison between the proposed autoencoder (AE) based receiver with that of the conventional receiver for different block sizes, noise correlation levels, and $h = 1$.

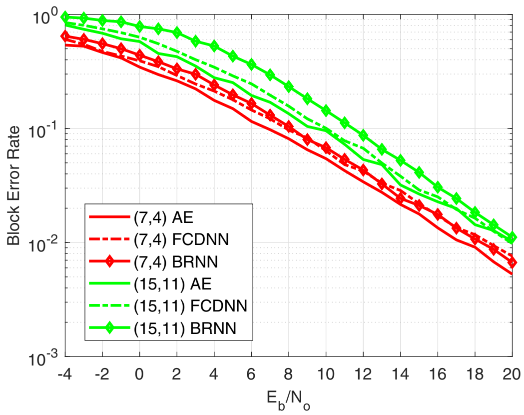


Fig. 5: BLER performance comparison between FCDNN based receiver, BRNN based receiver, and conventional receiver for a $(7,4)$ code and a $(15,11)$ code in Rayleigh fading channel.

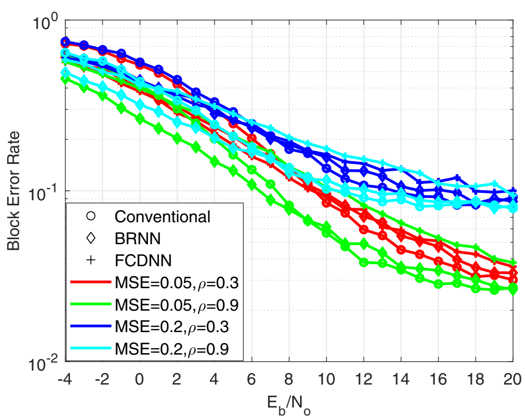


Fig. 6: BLER performance comparison between proposed BRNN based receiver and conventional receiver for (7,4) block code in Rayleigh fading channel for different levels of model mismatches.

perfect. A perturbation or mismatch in the channel model can lead to degradation in performance. In this work, we proposed an autoencoder based transceiver using DNN that are robust to channel model mismatches. We designed a loss function for the DNN which enables it to learn block codes with optimal distance properties to encode the input data. We proposed two receiver architectures based on FCDNN and BRNN for complex fading channels. We showed that the performance of the proposed DNN based transceivers is superior to that of the conventional receiver in the presence of practical channel mismatches such as imperfect CSI and noise correlation. Thus, we demonstrated that the proposed DNN based transceivers can be quite robust, and provide high reliability and immunity to channel mismatches in practical scenarios compared to conventional transceivers.

REFERENCES

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: trends, perspectives, and prospects," *Science magazine*, vol. 349, pp. 255-260, Jul. 2015.

[2] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognitive Commun. and Netw.*, vol. 3, pp. 563-575, Dec. 2017.

[3] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, M.I.T. Press, 1996.

[4] A. S. Hadi, "Linear block code decoder using neural network," *IEEE Intl. Joint Conf. on Neural Networks*, Sep. 2008.

[5] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, "LEARN codes: inventing low-latency codes via recurrent neural networks," *Proc. IEEE ICC'2019*, Jul. 2019.

[6] T. Wu, "CNN and RNN-based deep learning methods for digital signal demodulation," *Proc. ACM IVSP'2019*, Feb. 2019.

[7] A. Al-Baidhani and H. H. Fan, "Learning for detection: a deep learning wireless communication receiver over Rayleigh fading channels," *Proc. IEEE ICNC'2019*, Feb. 2019.

[8] V. Raj and S. Kalyani, "Backpropagating through the air: deep learning at physical layer without channel models," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2278-2281, Nov. 2018.

[9] B. Zhu, J. Wang, L. He, and J. Song, "Joint transceiver optimization for wireless communication PHY using neural network," *IEEE J. Sel. Areas in Commun.*, vol. 37, no. 6, pp. 1364-1373, Jun. 2019.

[10] M. S. Baek, D. Kwak, J. Y. Jung, H. Kim, and D. J. Choi, "Implementation methodologies of deep learning-based signal detection for conventional MIMO transmitters," *IEEE Trans. Broadcasting*, vol. 65, no. 3, pp. 636-642, Sep. 2019.

[11] A.Van Der Ziel, "Flicker noise in electronic devices," *Advances in Electronics and Electron Physics*, vol. 49, pp. 225-297, 1979.

[12] J. L. Tandon and H. R. Bilger, "$1/f$ noise as a nonstationary process: experimental evidence and some analytical conditions," *Journal of Applied Physics*, vol. 47, no. 4, pp. 1697-1701, Apr. 1976.

[13] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and Helmholtz free energy," *Proc. NIPS*, pp. 3-10, Nov. 1993.

[14] H. David and H. Sarah, *Digital Design and Computer Architecture*, 2007.

[15] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673-2681, Nov. 1997.

[16] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge Univ. Press, 2005.