Future Generation Computer Systems 95 (2019) 629–638

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Deep neural network-aided Gaussian message passing detection for ultra-reliable low-latency communications

Jie Guo [a], Bin Song [a,*], Yuhao Chi [a], Lahiru Jayasinghe [b], Chau Yuen [b], Yong Liang Guan [c], Xiaojiang Du [d], Mohsen Guizani [e]

[a] *State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, 710071, China*
[b] *Singapore University of Technology and Design, 487372, Singapore*
[c] *Nanyang Technological University, 639798, Singapore*
[d] *Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA*
[e] *Department of College of Engineering, Qatar University, Qatar*

HIGHLIGHTS

- The GMP algorithm with the aid of deep neural network (DNN).
- Combining the advantages of DNN and GMP, namely low-complexity and high-reliable.
- Improving the reliability in finite-length systems.
- DNN aids the GMP and SA-GMP algorithms to converge fast with less iteration.
- Robust to the cases with imperfect channel estimations.

ARTICLE INFO

Article history:
Received 28 July 2018
Received in revised form 23 December 2018
Accepted 21 January 2019
Available online 29 January 2019

Keywords:
URLLC
Deep neural network
Message passing
Signal recovery
Loopy factor graph

ABSTRACT

Ultra-reliable low-latency communications (URLLC) is a key technology in 5G supporting real-time multimedia services, which requires a low-cost signal recovery technology in the physical layer. A kind of well-known low-complexity signal detection is message passing algorithm (MPA) based on factor graph. However, reliability and robustness of MPA are deteriorated when there are cycles in factor graph. To address this issue, we propose two novel Gaussian message passing (GMP) algorithms with the aid of deep neural network (DNN), in which the network architectures consist of two DNNs associated with detections for mean and variance of the signal. Particularly, the network architecture is constructed by transforming the factor graph and message update functions of the original GMP algorithm from node-type into edge-type. Then, weights and bias parameters are assigned in the network architecture. With the aid of deep learning methods, the optimal weights and bias parameters are obtained. Numerical results demonstrate that two proposed DNN-aided GMP algorithms can significantly improve the convergence of original GMP algorithm and also achieve robust performances in the cases without prior information.

1. Introduction

Nowadays, with the rapid development of smart wireless devices, the future communication mechanisms consists of coexistence of human-centric and machine-type services as well as hybrids of these cases, which is very different from traditional human-centric communications [1]. To support such various applications, 5G wireless services are classified into three categories: ultra-reliable and low-latency communication (URLLC), enhanced mobile broadband (eMBB), and massive machine-type communication (mMTC), which includes vehicle-to-vehicle (V2V) communications [2], machine-to-machine (M2M) communications [3], Internet-of-things (IoT) [4–7], and cloud radio access network (C-RAN) [8–11].

Due to the fact that URLLC has two stringent requirements: low latency and high reliability, the design of physical-layer technologies is very challenging, in which signal recovery is the core technology to guarantee the successful transmissions of messages in wireless networks. Among signal detection algorithms, it is well known that Minimum Mean Square Error (MMSE) detection is optimal in the case of linear detection of Gaussian sources in noisy channels. Nevertheless, the complexity of MMSE detection is very high due to performing matrix inversion. In order to avoid the

* Corresponding author.
E-mail address: bsong@mail.xidian.edu.cn (B. Song).

https://doi.org/10.1016/j.future.2019.01.041
0167-739X/© 2019 Elsevier B.V. All rights reserved.

matrix inversion, one of popular low-complexity methods for signal recovery is message passing algorithm (MPA) based on factor graph [12–17], in which the optimal recovery for signals is decomposed into many distributed local calculations at nodes in the factor graph. The specific MPAs are required to be designed for different applications. For example, belief-propagation (BP) algorithm is designed for low-density parity-check (LDPC) decoding [17]. Gaussian message passing (GMP) are proposed for the massive multiple-input-multiple-output (MIMO) systems and the MIMO non-orthogonal multiple access (NOMA) systems respectively [18–20]. In [21,22], the MPAs are designed for mmWave MIMO and C-RAN systems respectively. Although the applications of MPAs are very extensive, the problem of convergence severely restricts the effectivity and robustness of the MPAs [17–22], which is still very intractable now. The reason is that the MPAs can converge to the optimal solutions in a tree-like factor graph [12,13], but easily diverge when there are cycles in the factor graph. To improve the convergences of the GMP algorithms, the scaled-and-added (SA) GMP algorithms are proposed in [18–20] based on the law of large numbers and infinite iteration number, which have been proved to be converged to the MMSE detector. However, the SA-GMP algorithms cannot also be always effective for the finite-length systems with finite iterations.

Recently, powerful deep neural networks (DNNs) [23] have been applied to the communications [24–32], whose results are superior/comparable to those of the conventional methods in communications. In [27], DNN is used to recover chemical signals in molecular communications, wherein precise mathematical models cannot represent the channel. Single-user MIMO systems based on DNN are proposed in [28], which achieve better performances than the conventional MIMO cases. In [29], DNN is employed for channel estimation and signal recovery in the orthogonal frequency-division multiplexing (OFDM) system, which is more robust to channel conditions than the conventional methods. Note that standard DNNs are used as black boxes commonly in [27–29]. Inspired by [30], a DNN can be designed by unfolding an existing iterative algorithm. Thus, a BP decoding method based on DNN is proposed to decode short binary channel codes [31] and a projected gradient descent method based on DNN is used to detect binary signals in MIMO systems [32], which are regarded as binary classification problems. However, the activation functions of these methods [31, 32] cannot be directly used for detection of real signals.

In this paper, a DNN-aided GMP algorithm is proposed to address the problem of signal recovery in URLLC systems. Specifically, a DNN is constructed explicitly by transforming the factor graph of original GMP from node-type into edge-type, which consists of two neural networks associated with detections for mean and variance of the signal. Then, activation functions at each layer are designed according to the message update functions of GMP. Weight and bias parameters are assigned in the DNN, which are trained with the aid of deep learning methods. In order to further achieve more reliable performance, a DNN-aided SA-GMP algorithm is proposed based on the original SA-GMP algorithm, which is designed similarly as the DNN-aided GMP algorithm. The major contributions of this paper are summarized as follows.

1. DNN is combined with GMP and SA-GMP algorithms to improve the reliability in finite-length systems.
2. DNN aids the GMP and SA-GMP algorithms to converge fast with less iteration.
3. Numerical results demonstrate that the proposed DNN-aided GMP and DNN-aided SA-GAMP algorithms are robust to the cases with imperfect channel estimations and the cases without priori information.

The rest of this paper is organized as follows. In Section 2, problem formulation for signal recovery is presented. In Section 3, the proposed neural network architecture and particular DNN-GMP algorithm are given. Section 4 provides a DNN-SA-GMP algorithm and Section 5 presents various simulations to validate the reliability and robustness of proposed DNN-aided MPA detection methods in URLLC systems. Finally, Section 6 concludes this paper and provides some future works.

2. Problem formulation

In this paper, we consider the recovery of signal vector $\boldsymbol{x} = [x_1, \ldots, x_K]^T$ from a noisy measurement $\boldsymbol{y} \in \mathcal{R}^{M\times 1}$:

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

where $\boldsymbol{H} \in \mathcal{R}^{M\times K}$ is a given measurement matrix and $\boldsymbol{n} = [n_1, \ldots, n_M]^T$ is an additive Gaussian noise vector obeying $\mathcal{N}(0, \sigma_n^2 \boldsymbol{I}_M)$ with an $M \times M$ identity matrix $\boldsymbol{I}_M$.

Note that we assume that entries of $\boldsymbol{x}$ obey independent Gaussian distributions, i.e., $x_k \sim \mathcal{N}(0, \sigma_k^2)$, $k = 1, \ldots, K$. Although discrete modulated signals are used in real communications, independent Gaussian sources are also usually employed in the design of communication network [33–36]. The reason is that in order to enable URLLC, the statistical distribution of transmitted discrete signals should be approximated as Gaussian distribution according to Shannon theory [37,38], which could employ Gallage mapping [37,38] or superposition coded modulation [39,40] to generate Gaussian-like transmit signals. Therefore, the proposed detection method based on Gaussian sources can be extended for discrete signals cases.

In order to recover Gaussian signal vector $\boldsymbol{x}$ with low complexity, the GMP algorithm is employed in [18–20]. As shown in Fig. 1, a pairwise factor graph for the GMP algorithm is presented, which consists of Gaussian nodes, variable nodes, sum nodes, noise nodes, and the corresponding edges. Based on the factor graph, we briefly introduce the GMP algorithm. Message update among nodes in the GMP is similar to the BP decoding for LDPC codes [17], but the differences from the BP decoding are Gaussian messages passing along edges and update functions for Gaussian messages at nodes. The GMP algorithm is given as follows.

2.1. Message update at sum nodes

In Fig. 1, each sum node is regarded as a multiple-access process, such that message update at the mth sum node is

$$\begin{cases} e^s_{m\to k}(t) = y_m - \sum\limits_{i\neq k} h_{mi} e^v_{i\to m}(t-1), \\ v^s_{m\to k}(t) = \sum\limits_{i\neq k} h^2_{mi} v^v_{i\to m}(t-1) + \sigma_n^2, \end{cases} \tag{2}$$

where $m \in \mathcal{M}, \mathcal{M} = \{1, \ldots, M\}, i, k \in \mathcal{K}, \mathcal{K} = \{1, \ldots, K\}, y_m$ is the mth entry of $\boldsymbol{y}$, h_{mi} is the entry in the mth row and the ith column of $\boldsymbol{H}$, and t denotes the iteration index. Let $e^v_{i\to m}$ and $v^v_{i\to m}$ denote the mean and variance passing from the ith variable node to the mth sum node. Let $e^s_{m\to k}$ and $v^s_{m\to k}$ denote the mean and variance passing from the mth sum node to the kth variable node. Initially, values of $e^v_{i\to m}(0)$ and $v^v_{i\to m}(0)$ equal to 0 and $+\infty$ respectively.

2.2. Message update at variable nodes

In Fig. 1, each variable node is regarded as a broadcast process, such that message update at the kth variable node is

$$\begin{cases} v^v_{k\to m}(t) = (\sum\limits_{j\in\mathcal{M}} h^2_{jk} v^s_{j\to k}(t)^{-1} + \sigma_k^{-2})^{-1}, \\ e^v_{k\to m}(t) = v^v_{k\to m}(t)(\sum\limits_{j\in\mathcal{M}} h_{jk} v^s_{i\to k}(t)^{-1} e^s_{j\to k}(t)), \end{cases} \tag{3}$$

where $k \in \mathcal{K}$ and $m, j \in \mathcal{M}$.

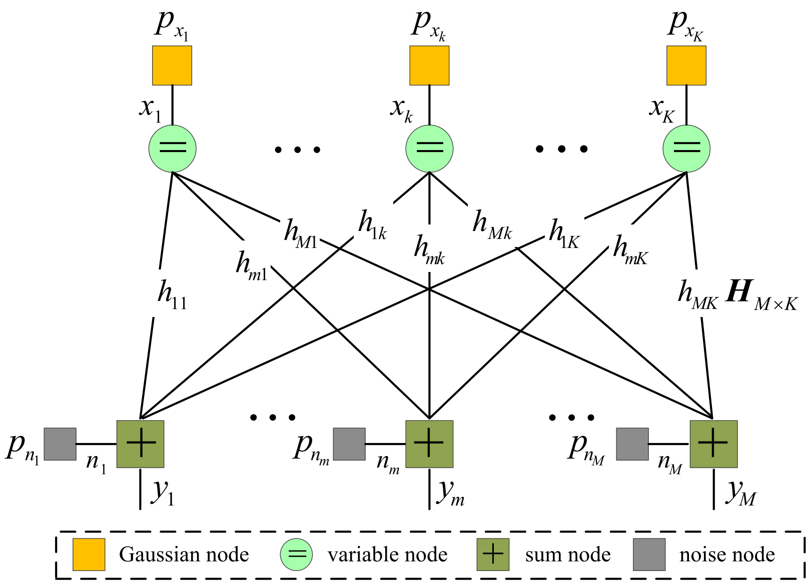


Fig. 1. Factor graph of GMP for signal recovery.

2.3. *Decision and output of GMP*

The iterative process between sum nodes and variable nodes will stop when the MSE requirement is satisfied or the preset maximum iteration number is reached. Estimated signal $\hat{x}_k$ and obtained variance $\sigma^2_{\hat{x}_k}$ are

$$\begin{cases} \sigma^2_{\hat{x}_k} = (\sum\limits_m h^2_{mk} v^s_{m\to k}(t)^{-1} + \sigma_k^{-2})^{-1}, \\ \hat{x}_k = \sigma^2_{\hat{x}_k}\big(\sum\limits_m h_{mk} v^s_{m\to k}(t)^{-1} e^s_{m\to k}(t)\big). \end{cases} \qquad (4)$$

However, as shown in Fig. 1, there are a large number of cycles in the factor graph, such that the GMP algorithm may easily diverge. As a result, the SA-GMP algorithm is proposed to improve the convergence of the GMP algorithm based on the law of large numbers [18–20]. Nonetheless, the SA-GMP algorithm does not always work well in finite-length systems. Therefore, our goal is to exploit the powerful DNN to aid the convergences of GMP and SA-GMP algorithms in finite-length systems.

3. Deep neutral network — GMP

In this section, we propose a DNN-GMP algorithm based on the constructed neural network below. With the aid of DNN [23], training the parameters of neural network is to enhance reliable messages and suppress unreliable messages properly for the original GMP algorithm during each iterative detection.

3.1. *Construction of neural network*

Note that a deep learning network can be designed by unfolding an existing iterative algorithm [30–32,41]. Similarly, to construct a DNN for GMP effectively, we transform the message update functions and factor graph of the original GMP from node-type into edge-type. Meanwhile, since the Gaussian signals are detected based on the estimations of means and variances in Eqs. (2)–(4), the proposed DNN-GMP consists of two neural networks associated with detections for means and variance respectively.

To be specific, let the maximum iteration number be L, and the total number of edges in the factor graph be $E = M \times K$. Through unfolding the message update functions in Eqs. (2)–(4), the both proposed neural networks of detections for means and variances consist of $2L + 2$ layers, which include one input layer, $2L$ hidden layers, and one output layer. The number of nodes in the input layer, each hidden layer, and the output layer is M, E, and K respectively. Then, the inputs of the neural networks associated with means and variances detections are received signal $\boldsymbol{y}$ and noise variance σ_n^2 respectively. The outputs of the output layers in the above two networks are estimated signal $\hat{\boldsymbol{x}} = [\hat{x}_1, \ldots, \hat{x}_K]^T$ and estimated variance $\boldsymbol{\sigma}^2_{\hat{\boldsymbol{x}}} = [\sigma^2_{\hat{x}_1}, \ldots, \sigma^2_{\hat{x}_K}]^T$ respectively. For hidden middle layers of the above two networks, the inputs are the means and variances of messages passing from the previous layer, and the outputs are the means and variances of updated messages passing to the next layer. In this way, the proposed neural network is obtained.

According to the original GMP algorithm, the activation functions and parameters of the DNN are designed to obtain the DNN-GMP algorithm. Due to the different rules of messages update at sum nodes and variable nodes, the activation functions in the even hidden layers and the odd hidden layers employ the message update functions at sum nodes and variable nodes respectively. The activation functions in the input layers are linear functions with respect to $\boldsymbol{y}$ and σ_n^2. The activation functions in the output layers are the combination of full estimated messages associated with mean and variances respectively. Then, weight parameters are assigned to the messages on all edges among layers of the DNNs for mean and variance detections. Bias parameters are assigned to all edges among layers of the DNN for variance detection. The initial values of these weight and bias parameters are 1.

To illustrate the DNN architecture clearly, we take an example, where $K = 3, M = 4$, and $L = 2$. The DNN architecture is shown in Fig. 2, where the indices of nodes in the hidden layers denote those of edges in the factor graph. Here, the corresponding edges matrix $\mathcal{E}$ and edge-type channel matrix $\boldsymbol{H}^{\mathcal{E}}_{K\times M}$ are given as

$$\mathcal{E} = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \\ e_{10} & e_{11} & e_{12} \end{bmatrix},$$

$$\boldsymbol{H}^{\mathcal{E}}_{K\times M} = [h_{11}\mathbf{1}_{K\times M}\ h_{12}\mathbf{1}_{K\times M}\ h_{1K}\mathbf{1}_{K\times M}\ \ldots\ h_{MK}\mathbf{1}_{K\times M}],$$

where $\mathbf{1}_{K\times M}$ denotes an all-ones column vector of length $K \times M$.

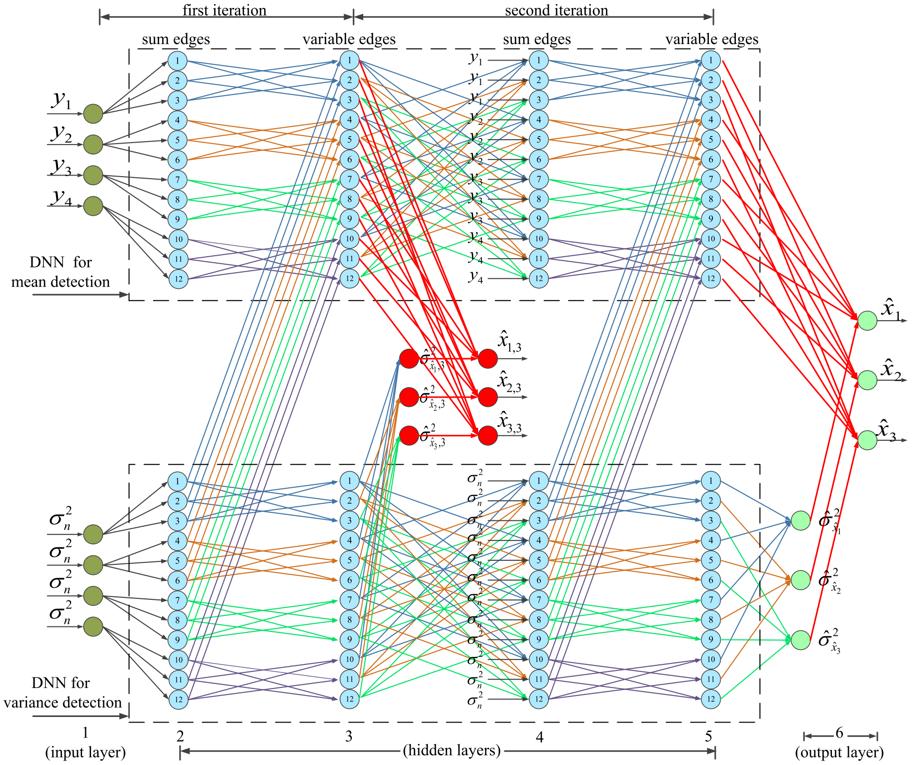


Fig. 2. Network architecture of the DNN-GMP for signal recovery ($K = 3, M = 4, L = 2$).

3.2. DNN-GMP algorithm

Based on the constructed DNN architecture, the DNN-GMP algorithm is proposed, whose differences from the original GMP algorithm are the edge-type message update functions and training network parameters in the DNN. The update process is given as follows in detail.

3.2.1. Message update in even hidden layers

$$\begin{cases} e^s_{\ell_1,d=(m,k)} &= w^{y_m}_{\ell_1,d} y_m - \sum\limits_{d'=(i,m),i\neq k} w^{e^s}_{\ell_1,d,d'} h_{mi} e^v_{\ell_1-1,\,d'}, \\ v^s_{\ell_1,d=(m,k)} &= \sum\limits_{d'=(i,m),i\neq k} w^{v^s}_{\ell_1,d,d'} h^2_{mi} v^v_{\ell_1-1,d'} + w^n_{\ell_1,d}\sigma^2_n + b^n_{\ell_1,d}, \end{cases}$$

where $m \in \mathcal{M}$, $k, i \in \mathcal{K}$, $d, d' \in \mathcal{E}' = \{1, \ldots, E\}$, $\ell_1 \in \mathcal{L} = \{2, \ldots, 2L + 1\}$, ℓ_1 is the index of even hidden layers, $e^s_{\ell_1,d=(m,k)}$ and $v^s_{\ell_1,d=(m,k)}$ denote the mean and variance passing from the mth sum node in the ℓ_1th layer to the kth variable node in the (ℓ_1+1)th layer, y_m is the mth entry of $\boldsymbol{y}$, h_{mi} is the entry in the mth row and the ith column of $\boldsymbol{H}$, $e^v_{\ell_1-1,d'=(i,m)}$ and $v^v_{\ell_1-1,d'=(i,m)}$ denote the mean and variance passing from the ith variable node in the $(\ell_1 - 1)$th layer to the mth sum node in the ℓ_1th layer, $b^n_{\ell_1,d}$ denotes the bias parameter, and $\{w^{y_m}_{\ell_1,d}, w^{e^s}_{\ell_1,d,d'}, w^{v^s}_{\ell_1,d,d'}\}$ denotes weights on the edges between the $(\ell_1 - 1)$th layer and the ℓ_1th layer. The initial values of $e^v_{1,d'}$ and $v^v_{1,d'}$ are 0 and $+\infty$ respectively.

3.2.2. Message update in odd hidden layers

$$\begin{cases} v^v_{\ell_2,d=(k,m)} &= (\sum\limits_{d'=(j,k),j\neq m} w^{v^v}_{\ell_2,d,d'} h^2_{jk} v^{s^{-1}}_{\ell_2-1,d'} + w^k_{\ell_2,d}\sigma_k^{-2} \\ &\quad +b^k_{\ell_2,d})^{-1}_{,} \\ e^v_{\ell_2,d=(k,m)} &= v^v_{\ell_2,d=(k,m)} \sum\limits_{d'=(j,k),j\neq m} w^{e^v}_{\ell_2,d,d'} h_{jk} v^{s^{-1}}_{\ell_2-1,d'} e^s_{\ell_2-1,d'}, \end{cases}$$

where $k \in \mathcal{K}, m, j \in \mathcal{M}, d, d' \in \mathcal{E}', b^k_{\ell_2,d}$ denotes the bias parameter, $\{w^{e^v}_{\ell_2,d,d'}, w^{v^v}_{\ell_2,d,d'}\}$ denotes weights on the edges between the $(\ell_2 - 1)$th layer and the ℓ_2th layer, and ℓ_2 takes odd values in $\mathcal{L}$.

3.2.3. Message combination in odd hidden layers

$$\begin{cases} \sigma^2_{\hat{x}_k,\ell_2} &= (\sum\limits_{d=(m,k)} h^2_{mk} v^{s^{-1}}_{\ell_2-1,d} + \sigma_k^{-2})^{-1}_{,} \\ \hat{x}_{k,\ell_2} &= \sigma^2_{\hat{x}_k}(\sum\limits_{d=(m,k)} h_{mk} v^{s^{-1}}_{\ell_2-1,d} e^s_{\ell_2-1,d}), \end{cases}$$

where $\hat{x}_{k,\ell_2}$ and $\sigma^2_{\hat{x}_k,\ell_2}$ are the estimated means and variances of users' signals at the output of the odd hidden layers. $\hat{x}_{k,\ell_2}$ denotes the estimated signal when one iteration detection is finished, such that the accuracy of detection in each iteration can be traced by taking into account the MSE between $\hat{x}_{k,\ell_2}$ and true signal $\boldsymbol{x}$ in the loss function.

3.2.4. Message combination in the output layer

$$\begin{cases} \sigma^2_{\hat{x}_k} &= (\sum\limits_{d=(m,k)} w^v_{2L+1,d} h^2_{mk} v^{s^{-1}}_{2L+1,d} + w^k_{2L+1,d}\sigma_k^{-2} + b^k_{2L+1,d})^{-1}_{,} \\ \hat{x}_k &= \sigma^2_{\hat{x}_k}(\sum\limits_{d=(m,k)} w^e_{2L+1,d} h_{mk} v^{s^{-1}}_{2L+1,d} e^s_{2L+1,d}), \end{cases}$$

where $k \in \mathcal{K}$, $m \in \mathcal{M}$, $d \in \mathcal{E}'$, $b^k_{2L+1,d}$ denotes the bias parameter, $\{w^e_{2L+1,d}, w^v_{2L+1,d}\}$ denotes weights on the edges between the last hidden layer and the output layer, and $\hat{x}_k$ is the kth element of estimated signal $\hat{\boldsymbol{x}}$.

3.3. DNN-GMP in matrix form

Note: let $\boldsymbol{H}^{\mathcal{E},2}_{K\times M} = \boldsymbol{H}^{\mathcal{E}}_{K\times M} \bullet \boldsymbol{H}^{\mathcal{E}}_{K\times M}$, $\bullet$ denote Hadamard product, $\boldsymbol{\sigma^2_n} = [\sigma^2_n]_{E\times 1}, \boldsymbol{\sigma^2_x} = [\sigma^2_{x_{d,k}}]_{E\times 1}, \boldsymbol{W}^{in}_y = [w^{y_m}_{in,d}]_{E\times M}, \boldsymbol{W}^{in}_e = [w^{e^s}_{in,d}]_{E\times E},$

$\boldsymbol{W}_v^{in} = [w_{in,d}^{v^s}]_{E\times E}$, $\boldsymbol{W}_n^{in} = [w_{in,d}^{n}]_{E\times 1}$, $\boldsymbol{B}_n^{in} = [b_{in,d}^{n}]_{E\times 1}$, $\boldsymbol{E}_v^{in} = [e_{in,d'}^{v}]_{E\times 1}$, $\boldsymbol{V}_v^{in} = [v_{in,d'}^{v}]_{E\times 1}$, $\boldsymbol{W}_y^{\ell_1} = [w_{\ell_1,d}^{y_m}]_{E\times M}$, $\boldsymbol{W}_e^{\ell_1} = [w_{\ell_1,d,d'}^{e^s}]_{E\times E}$, $\boldsymbol{W}_v^{\ell_1} = [w_{\ell_1,d,d'}^{v^s}]_{E\times E}$, $\boldsymbol{W}_n^{\ell_1} = [w_{\ell_1,d}^{n}]_{E\times 1}$, $\boldsymbol{B}_n^{\ell_1} = [b_{\ell_1,d}^{n}]_{E\times 1}$, $\boldsymbol{W}_e^{\ell_2} = [w_{\ell_2,d,d'}^{e^v}]_{E\times E}$, $\boldsymbol{W}_v^{\ell_2} = [w_{\ell_2,d,d'}^{v^v}]_{E\times E}$, $\boldsymbol{W}_{\boldsymbol{x}}^{\ell_2} = [w_{\ell_2,d}^{k}]_{E\times 1}$, $\boldsymbol{B}_{\boldsymbol{x}}^{\ell_2} = [b_{\ell_2,d}^{k}]_{E\times 1}$, $\boldsymbol{W}_e^{out} = [w_{2L+1,d}^{e}]_{K\times E}$, $\boldsymbol{W}_{\boldsymbol{x}}^{out} = [w_{\ell,d}^{k}]_{K\times 1}$, $\boldsymbol{W}_v^{out} = [w_{2L+1,d}^{v}]_{K\times E}$, $\boldsymbol{\sigma}_{\hat{\boldsymbol{x}}}^{\boldsymbol{2}} = [\sigma_{\hat{x}_{d,k}}^2]_{K\times 1}$, $\boldsymbol{B}_{\boldsymbol{x}}^{out} = [b_{\ell_2,d}^{k}]_{K\times 1}$. $\boldsymbol{E}_{sv}^{\ell_1\to\ell_2}$ and $\boldsymbol{V}_{sv}^{\ell_1\to\ell_2}$ denote the estimated mean and variance passing from the ℓ_1-th layers to the ℓ_2-th layers respectively. $\boldsymbol{E}_{vs}^{\ell_2\to\ell_1}$ and $\boldsymbol{V}_{vs}^{\ell_2\to\ell_1}$ denote the estimated mean and variance passing from the ℓ_2-th layers to the ℓ_1-th layers respectively $\boldsymbol{E}_{\hat{\boldsymbol{x}},\ell_2}$ and $\boldsymbol{V}_{\boldsymbol{\sigma}_{\hat{\boldsymbol{x}}}^2,\ell_2}$ denote the message combination in the odd hidden layers respectively. $\boldsymbol{E}_{\hat{\boldsymbol{x}}}$ and $\boldsymbol{V}_{\boldsymbol{\sigma}_{\hat{\boldsymbol{x}}}^2}$ denote the message combination in the output layers respectively. As shown in Fig. 2, the proposed DNN is not fully connected so that all weight matrices consisting of nonzero elements and zeros are designed according to message update rules in Eqs. (2)–(4). Algorithm 1 shows the detailed process of matrix-form DNN-GMP.

Algorithm 1 DNN-GMP Algorithm

```
Input: $\boldsymbol{y}, \boldsymbol{H}_{K\times M}^{\mathcal{E}}, \boldsymbol{H}_{K\times M}^{\mathcal{E},2}, \boldsymbol{H}_{K}^{\mathcal{E}}, \boldsymbol{H}_{K}^{\mathcal{E},2}, \boldsymbol{\sigma}_{\boldsymbol{x}}^{\boldsymbol{2}}, \boldsymbol{\sigma}_{\hat{\boldsymbol{x}}}^{\boldsymbol{2}}, \boldsymbol{\sigma}_n$.
Initialization: $\boldsymbol{E}_v^{in} = \boldsymbol{0}, \boldsymbol{V}_v^{in} = +\infty$.
Input layer $\to$ $\ell_1 = 2$ even hidden layer:
    $\boldsymbol{E}_{sv}^{\ell_1=2\to\ell_2=3} = \boldsymbol{W}_y^{in}\boldsymbol{y} - (\boldsymbol{W}_e^{in} \bullet \boldsymbol{H}_{K\times M}^{\mathcal{E}})\boldsymbol{E}_v^{in}$
    $\boldsymbol{V}_{sv}^{\ell_1=2\to\ell_2=3} = (\boldsymbol{W}_v^{in} \bullet \boldsymbol{H}_{K\times M}^{\mathcal{E},2})\boldsymbol{V}_v^{in} + \boldsymbol{W}_n^{in} \bullet \boldsymbol{\sigma}_{\boldsymbol{n}}^{\boldsymbol{2}} + \boldsymbol{B}_n^{in}$
Hidden layers:
for: $\ell_2 \in \mathcal{L}$ odd hidden layer:
    $\boldsymbol{V}_{vs}^{\ell_2\to\ell_1} = [(\boldsymbol{W}_v^{\ell_2} \bullet \boldsymbol{H}_{K\times M}^{\mathcal{E},2})(\boldsymbol{V}_{sv}^{\ell_1\to\ell_2})^{-1} + \boldsymbol{W}_{\boldsymbol{x}}^{\ell_2} \bullet \boldsymbol{\sigma}_{\boldsymbol{x}}^{\boldsymbol{-2}} + \boldsymbol{B}_{\boldsymbol{x}}^{\ell_2}]^{-1}$
    $\boldsymbol{E}_{vs}^{\ell_2\to\ell_1} = \boldsymbol{V}_{vs}^{\ell_2\to\ell_1} \bullet [(\boldsymbol{W}_e^{\ell_2} \bullet \boldsymbol{H}_{K\times M}^{\mathcal{E}})((\boldsymbol{V}_{sv}^{\ell_1\to\ell_2})^{-1} \bullet \boldsymbol{E}_{sv}^{\ell_1\to\ell_2})]$
    $\boldsymbol{V}_{\boldsymbol{\sigma}_{\hat{\boldsymbol{x}}}^2,\ell_2} = [\boldsymbol{H}_K^{\mathcal{E},2}(\boldsymbol{V}_{sv}^{\ell_1\to\ell_2})^{-1} + \boldsymbol{\sigma}_{\hat{\boldsymbol{x}}}^{\boldsymbol{2}}]^{-1}$
    $\boldsymbol{E}_{\hat{\boldsymbol{x}},\ell_2} = \boldsymbol{V}_{\boldsymbol{\sigma}_{\hat{\boldsymbol{x}}}^2,\ell_2} \bullet [\boldsymbol{H}_K^{\mathcal{E}}((\boldsymbol{V}_{sv}^{\ell_1\to\ell_2})^{-1} \bullet \boldsymbol{E}_{sv}^{\ell_1\to\ell_2})]$
for: $\ell_1 \in \mathcal{L}$ even hidden layer:
    $\boldsymbol{E}_{sv}^{\ell_1\to\ell_2} = \boldsymbol{W}_y^{\ell_1}\boldsymbol{y} - (\boldsymbol{W}_e^{\ell_1} \bullet \boldsymbol{H}_{K\times M}^{\mathcal{E}})\boldsymbol{E}_{vs}^{\ell_2\to\ell_1}$
    $\boldsymbol{V}_{sv}^{\ell_1\to\ell_2} = (\boldsymbol{W}_v^{\ell_1} \bullet \boldsymbol{H}_{K\times M}^{\mathcal{E},2})\boldsymbol{V}_{vs}^{\ell_2\to\ell_1} + \boldsymbol{W}_n^{\ell_1} \bullet \boldsymbol{\sigma}_{\boldsymbol{n}}^{\boldsymbol{2}} + \boldsymbol{B}_n^{\ell_1}$
Output layers:
    $\boldsymbol{V}_{\boldsymbol{\sigma}_{\hat{\boldsymbol{x}}}^2} = [(\boldsymbol{W}_v^{out} \bullet \boldsymbol{H}_K^{\mathcal{E},2})(\boldsymbol{V}_{sv}^{\ell_1=2L\to\ell_2=2L+1})^{-1} + \boldsymbol{W}_{\boldsymbol{x}}^{out} \bullet \boldsymbol{\sigma}_{\hat{\boldsymbol{x}}}^{\boldsymbol{2}} + \boldsymbol{B}_{\boldsymbol{x}}^{out}]^{-1}$
    $\boldsymbol{E}_{\hat{\boldsymbol{x}}} = \boldsymbol{V}_{\boldsymbol{\sigma}_{\hat{\boldsymbol{x}}}^2} \bullet [(\boldsymbol{W}_e^{out} \bullet \boldsymbol{H}_K^{\mathcal{E}})((\boldsymbol{V}_{sv}^{\ell_1=2L\to\ell_2=2L+1})^{-1} \bullet \boldsymbol{E}_{sv}^{\ell_1=2L\to\ell_2=2L+1})]$
```

3.4. Loss function

In order to ensure the high reliability of signal recovery, the goal is to train all weight matrices $\boldsymbol{W}_{\mathcal{E}} = \{\boldsymbol{W}_y^{in}, \boldsymbol{W}_e^{in}, \boldsymbol{W}_v^{in}, \boldsymbol{W}_n^{in}, \boldsymbol{W}_y^{\ell_1}, \boldsymbol{W}_e^{\ell_1}, \boldsymbol{W}_v^{\ell_1}, \boldsymbol{W}_n^{\ell_1}, \boldsymbol{W}_e^{\ell_2}, \boldsymbol{W}_v^{\ell_2}, \boldsymbol{W}_{\boldsymbol{x}}^{\ell_2}, \boldsymbol{W}_e^{out}, \boldsymbol{W}_v^{out}, \boldsymbol{W}_{\boldsymbol{x}}^{out}\}$ and bias $\boldsymbol{B}_{\mathcal{E}} = \{\boldsymbol{B}_n^{in}, \boldsymbol{B}_n^{\ell_1}, \boldsymbol{B}_{\boldsymbol{x}}^{\ell_2}, \boldsymbol{B}_{\boldsymbol{x}}^{out}\}$, $\ell_1, \ell_2 \in \mathcal{L}$, to achieve the minimum MSE. Here we present two kinds of loss function to train the DNN. One is termed as single loss, i.e.,

$$\min_{\{\boldsymbol{W}_{\mathcal{E}},\boldsymbol{B}_{\mathcal{E}}\}} \frac{1}{2}\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2,$$

which is in fact used for the end-to-end learning process. The other one is termed as multi-loss, i.e,

$$\min_{\{\boldsymbol{W}_{\mathcal{E}},\boldsymbol{B}_{\mathcal{E}}\}} \frac{1}{2}\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2 + \frac{1}{2}\sum_{\ell_2} \|\boldsymbol{x} - \hat{\boldsymbol{x}}_{\ell_2}\|_2^2,$$

which combines the effect of each odd hidden layer to strengthen each iterative detection actually.

The item $\|\boldsymbol{x} - \hat{\boldsymbol{x}}_{\ell_2}\|_2^2$ denotes the MSE between true signal $\boldsymbol{x}$ and estimated signal $\hat{\boldsymbol{x}}_{\ell_2}$ at the ℓ_2-th iteration, where $\ell_2 < L$. As a result, the second item $\sum_{\ell_2} \|\boldsymbol{x} - \hat{\boldsymbol{x}}_{\ell_2}\|_2^2$ of the multi-loss function is introduced to enforce the reliability of estimation at each iteration during the training process as soon as possible.

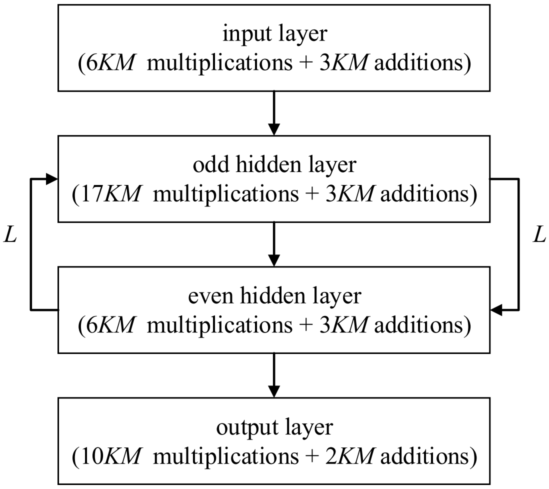

input layer
(6KM multiplications + 3KM additions)
odd hidden layer
(17KM multiplications + 3KM additions)
even hidden layer
(6KM multiplications + 3KM additions)
output layer
(10KM multiplications + 2KM additions)
L
L

Fig. 3. Computational complexity of each layer of the DNN-GMP algorithm.

Table 1
Complexity comparisons of IF, MMSE, GMP, and the proposed DNN-GMP detection algorithms.

Detection	IF	MMSE	GMP	DNN-GMP
Complexity	$\mathcal{O}(M^2K+M^3)$	$\mathcal{O}(min(MK^2+M^3, KM^2+M^3))$	$\mathcal{O}(MKL)$	$\mathcal{O}(MKL)$

To further investigate the effect of estimations at hidden layer, we also consider a loss function termed as layer loss, i.e.,

$$\min_{\{\boldsymbol{W}_{\mathcal{E}},\boldsymbol{B}_{\mathcal{E}}\}} \frac{1}{2}\sum_{\ell_2} \|\boldsymbol{x} - \hat{\boldsymbol{x}}_{\ell_2}\|_2^2,$$

where ℓ_2 takes an odd value and $\ell_2 < L$.

3.5. Complexity comparison

In our proposed DNN-GMP algorithm, although many weight and bias parameters are introduced, the neural network is trained offline on a HP Z840 workstation with NVIDIA GeForce GTX 1080 Ti. Consequently, the trained neural network can be stored in the on-device memory for online use. Therefore, the training complexity of DNN-GMP algorithm is just an offline complexity, such that the computation of DNN-GMP algorithm is dominated by the online calculation. According to Algorithm 1, the online computation complexities of input layer, each hidden layer, and output layer of DNN are calculated as shown in Fig. 3, which includes the number of multiplications and additions. Note that there are $\mathcal{O}(KM)$ multiplications and additions in each layer. As a result, the online complexity of our proposed DNN-GMP algorithm is as low as $\mathcal{O}(MKL)$, which is the same as the original GMP algorithm.

Among existing signal detection algorithms, it is well known the MMSE detection is optimal when users employ Gaussian sources. But the complexity of MMSE detection is very high due to performing matrix inversion operations [42]. On the other hand, inverse filter (IF) [43] is also known as zero-forcing or decorrelator receiver, which also needs to perform matrix inversion operations. In the GMP algorithm [18–20], since it is performed based on the factor graph, it can achieve linear complexity with K and M. Here we give the detailed complexity comparisons of the above algorithms in Table 1.

4. Deep neutral network — SA-GMP

Considering that the SA-GMP has been presented in [18,19] to improve the convergence of the GMP based on the law of large numbers, a DNN-SA-GMP is proposed to further improve the reliability of the SA-GMP in finite-length systems. Moreover, since the SA-GMP provides a more robust initialization of neural network than the GMP, the DNN-SA-GMP can take less training epochs to achieve the optimal network parameters. The detailed DNN-SA-GMP is given as follows.

4.1. *Message update in even hidden layers*

$$\begin{cases} e^{s}_{\ell_1,d=(m,k)} = w^{y'_m}_{\ell_1,d} y'_m - \sum\limits_{d'=(i,m),i\neq k} w^{e^s}_{\ell_1,d,d'} h'_{mi} e^{v}_{\ell_1-1,\,d'}, \\ v^{s}_{\ell_1,d=(m,k)} = \sum\limits_{d'=(i,m),i\neq k} w^{v^s}_{\ell_1,d,d'} h^2_{mi} v^{v}_{\ell_1-1,d'} + w_{\ell_1,d,n}\sigma^2_n \\ \qquad +b_{\ell_1,d,n}, \end{cases}$$

where $y' = \sqrt{\alpha}y$, $h'_{mi} = \sqrt{\alpha}h_{mi}$, and α is a relaxation parameter obtained from [18,19].

4.2. *Message update in odd hidden layers*

$$\begin{cases} v^{v}_{\ell_2,d=(k,m)} = (\sum\limits_{d'=(j,k)} w^{v^v}_{\ell_2,d,d'} h^2_{jk} v^{s^{-1}}_{\ell_2-1,d'} + w_{\ell_2,d,k}\sigma_k^{-2} \\ \qquad +b_{\ell_2,d,k})^{-1}_{,} \\ e^{v}_{\ell_2,d=(k,m)} = v^{v}_{\ell_2,d=(k,m)}(\sum\limits_{d'=(j,k)} w^{e^v}_{\ell_2,d,d'} h'_{jk} v^{s^{-1}}_{\ell_2-1,d'} e^{s}_{\ell_2-1,d'}) \\ \qquad -w^{e^v}_{\ell_2,d,d''}(\alpha-1)e^{v}_{\ell_2-2,d''=(k,m)}, \end{cases}$$

where $e^{v}_{\ell_2-2,d''=(k,m)} = 0$ for $\ell_2 \leq 2$, $\ell_2 \in \mathcal{L}$, and $d'' \in \mathcal{E}'$.

4.3. *Message combination in the output layer*

$$\begin{cases} \sigma^2_{\hat{x}_k} = (\sum\limits_{d=(m,k)} w^{v}_{2L+1,d} h^2_{mk} v^{s^{-1}}_{2L+1,d} + w^{k}_{2L+1,d}\sigma_k^{-2} + b^{k}_{2L+1,d})^{-1}_{,} \\ \hat{x}_k = \sigma^2_{\hat{x}_k}(\sum\limits_{d=(m,k)} w^{e}_{2L+1,d} h'_{mk} v^{s^{-1}}_{2L+1,d} e^{s}_{2L+1,d}) \\ \qquad -\frac{\alpha-1}{M} \sum\limits_{d''=(m,k)} w^{e^v}_{2L+1,d,d''} e^{v}_{2L-1,d''}, \end{cases}$$

Note that the online complexity of the DNN-SA-GMP is as low as $\mathcal{O}(MKL)$, which is same as the original GMP. Moreover, the training process of the DNN-SA-GMP is similar as that of the DNN-GMP.

5. Numerical results

In simulations, we consider the signal recovery in an uplink MIMO-NOMA system [44–46], which is the key multiple-access technology in 5G supporting URLLC systems. We assume that there are $K = 10$ single-antenna users and a base station equipped with $M = 20$ receive antennas. Here, we consider that entries of $\boldsymbol{x}$ obey independent and identically distributed (i.i.d.) Gaussian distribution $\mathcal{N}(0, 1)$, i.e., $\sigma_k^2 = 1$, $k \in \mathcal{K}$, and those of channel matrix $\boldsymbol{H}$ obey i.i.d. Gaussian distribution $\mathcal{N}(0, 1)$.

The training and testing datasets are generated independently according to Eq. (1) ($\boldsymbol{y} = \boldsymbol{Hx} + \boldsymbol{n}$) respectively. That is, input labels $\boldsymbol{x}$ and $\boldsymbol{H}$ are generated independently according to $\mathcal{N}(0, 1)$ and output label $\boldsymbol{y}$ is obtained based on $\boldsymbol{y} = \boldsymbol{Hx} + \boldsymbol{n}$, where $\boldsymbol{n}$ is generated according to $\mathcal{N}(0, \sigma_n^2)$. Note that signal-to-noise ratio (SNR) $= \frac{\|\boldsymbol{H}\|_2^2}{\sigma_n^2}$. Therefore, different output labels $\boldsymbol{y}$ are obtained under different SNRs. The training of the proposed DNN is implemented in TensorFlow [47] and conducted using stochastic gradient descent with mini-batch learning. In our experiments, we do not observe overfitting phenomenon. Details about our experiments and results are provided as follows.

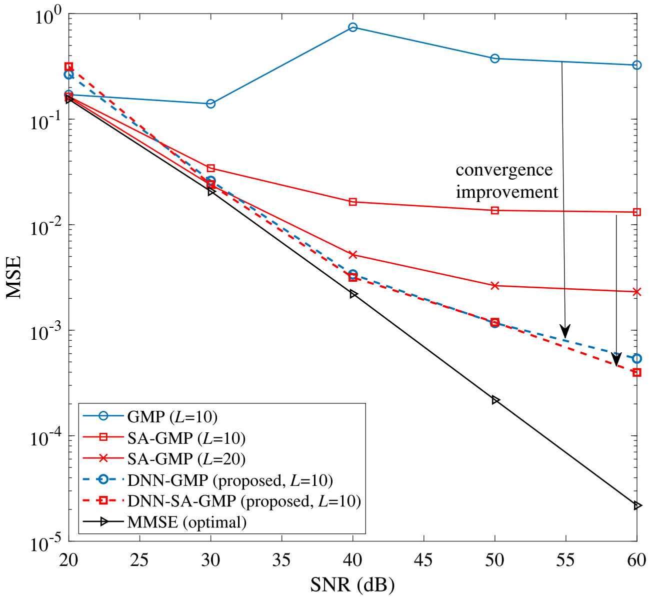


Fig. 4. MSE comparisons among the proposed DNN-GMP, the proposed DNN-SA-GMP, GMP [18,19], SA-GMP [18,19], and MMSE (optimal), where $K = 10$, $M = 20$, and $L \in \{10, 20\}$.

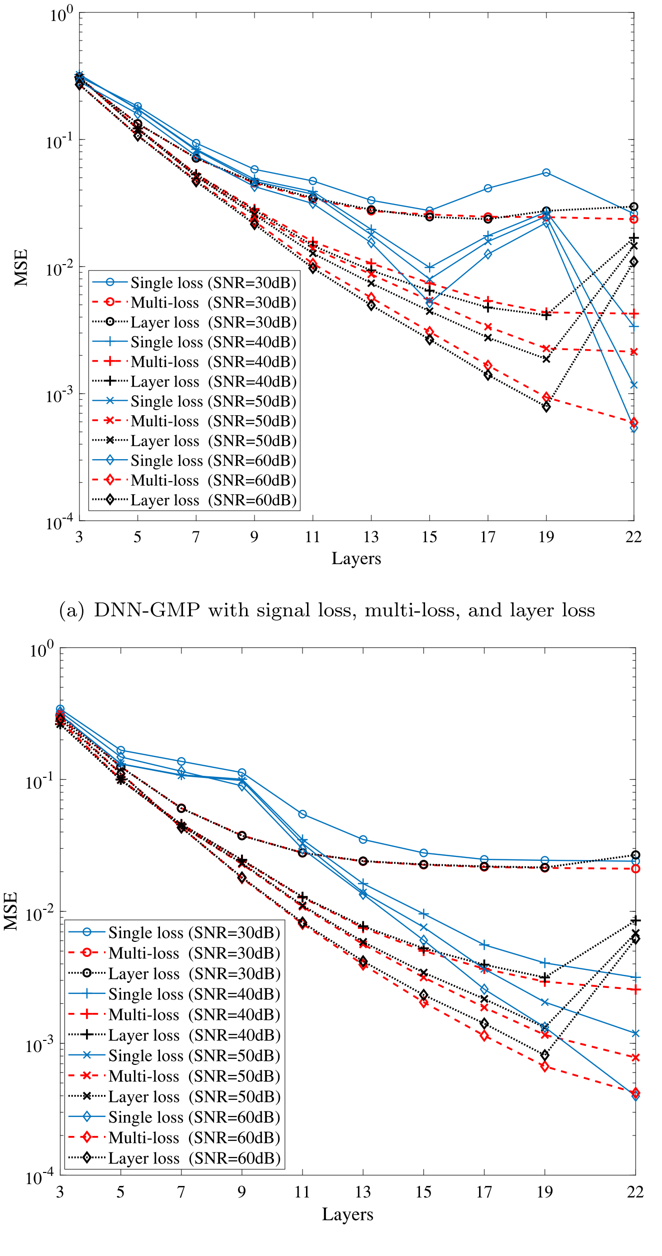
Let the maximum iteration number $L = 10$ and average MSE $= \frac{1}{K}E[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2]$. The size of training data is 100000, the size of mini-batch learning is 100, and the learning rate is adjusted gradually. The weight and bias parameters are trained under SNR = 60 dB and employed for each SNR ∈ {20 dB, 30 dB, 40 dB, 50 dB, 60 dB}. The online simulated MSEs are averaged over 100 realizations.

5.1. *MSE performance comparison*

To evaluate recovery accuracy of the proposed DNN-GMP and DNN-SA-GMP, we provide the MSE comparisons of the GMP [18,19], the SA-GMP [18,19], the DNN-GMP, the DNN-SA-GMP, and MMSE. As shown in Fig. 4, the MSEs of DNN-GMP and DNN-SA-GMP are more close to that of MMSE than those of GMP and SA-GMP over the almost entire SNR region, in which MMSE is optimal when employing Gaussian signals. Note that the MSE curve of GMP slightly diverges and that of SA-GMP converges to a bad MSE at 2×10^{-2}, the DNN-GMP and the DNN-SA-GMP converge more reliably with the aid of DNN. When the maximum iteration number L increases to 20, the GMP becomes diverging severely that is not given, but the SA-GMP becomes better. Nonetheless, our proposed DNN-GMP and DNN-SA-GMP still have large performance gains over the SA-GMP when SNR > 40 dB.

5.2. *MSE evolution in DNN*

Since L is set as 10, there are $2L + 2 = 22$ layers in the network architecture. In the proposed DNN-GMP and DNN-SA-GMP, the estimated signals can be traced at the outputs of the odd hidden layers and the output layer. Thus, Fig. 5 shows that the MSE performances of the DNN-GMP and DNN-SA-GMP evolve with the increase of layer number. In Fig. 5, we consider the effect of three loss functions on the MSE performances, i.e., signal loss,



(a) DNN-GMP with signal loss, multi-loss, and layer loss

(b) DNN-SA-GMP with signal loss, multi-loss, and layer loss

Fig. 5. Output MSEs of the odd hidden layers and the output layer of the proposed DNN-GMP and DNN-SA-GMP with signal loss, multi-loss, and layer loss functions under SNR ∈ {30 dB, 40 dB, 50 dB, 60 dB}, where $K = 10$, $M = 20$, and $L = 10$.

multi-loss, and layer loss. As shown in Fig. 5(a), the multi-loss function can help the DNN-GMP converge more reliably in each odd hidden layer than the single loss function. The MSE performances with multi-loss function improve almost linearly with increasing number of layers. Note that the MSE curve of layer loss function is close to the multi-loss function among the hidden layers, but becomes bad at the output layer when the index of layer is 22. The reason is that the layer loss function does not include the network parameters of the output layer, such that these network parameters are not trained effectively.

The similar phenomenon is also observed in Fig. 5(b), which shows the effect of loss functions on the MSE performances of the DNN-SA-GMP algorithm. This also indicates that the number of layers in the DNN can be determined when the MSE requirement is given.

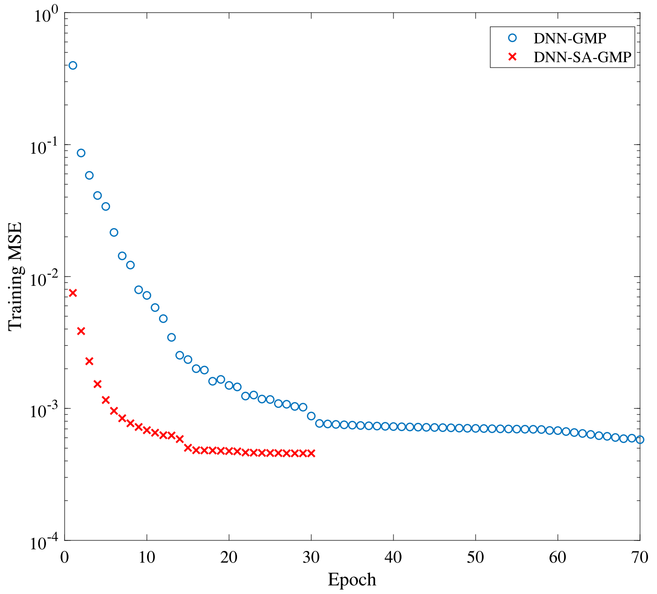


Fig. 6. MSE performances of the proposed DNN-GMP and DNN-SA-GMP algorithms with multi-loss function during the training process, where $K = 10$, $M = 20$, and $L = 10$.

5.3. *Convergence rate during training process*

To investigate the convergence rate of the proposed DNN-GMP and DNN-SA-GMP algorithms during the training process, we present the MSE curves under the multi-loss function for each training epoch as shown in Fig. 6. Note that the initial point of the DNN-SA-GMP is lower than that of the DNN-GMP due to the fact that the original SA-GMP algorithm provides a more robust initialization than the original GMP algorithm. Meanwhile, since the DNN-SA-GMP combines the advantages of SA-GMP and DNN at the same time, the DNN-SA-GMP converges faster than the DNN-GMP during the training process.

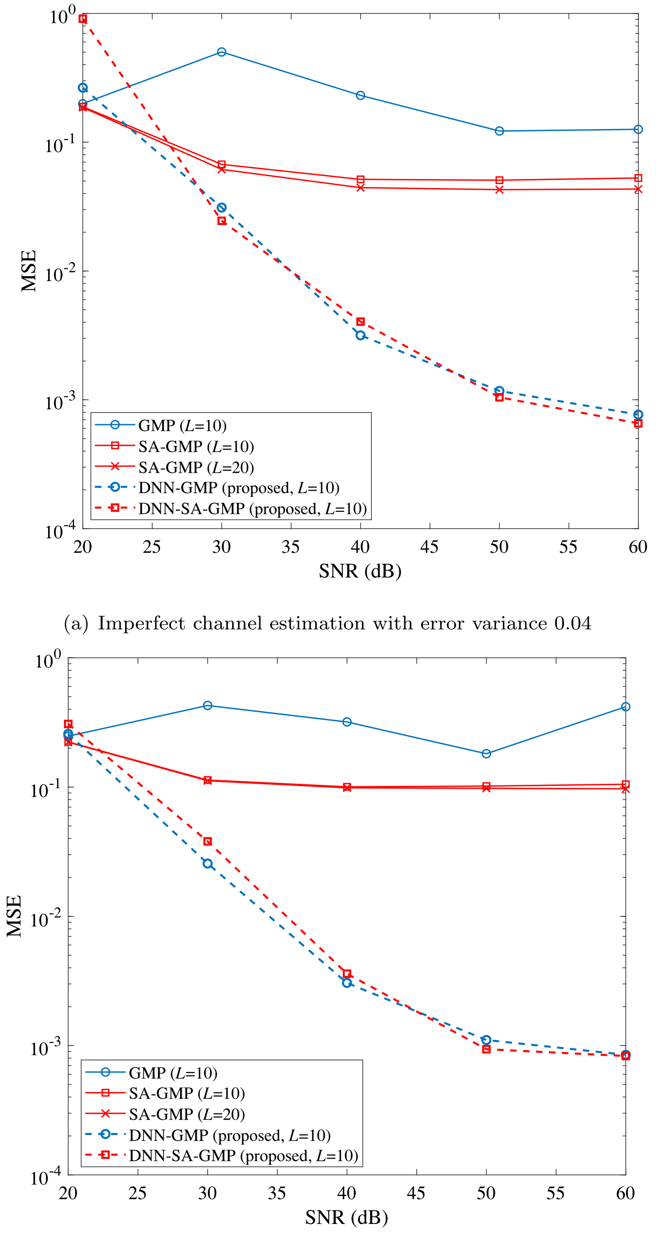
5.4. *Impact with imperfect channel estimation*

Since channel estimation is hard to be always estimated precisely, we consider the impact of imperfect channel estimation on the MSE performances of the proposed DNN-GMP and DNN-SA-GMP algorithms. Here we consider the variances of estimated channel errors are 0.04 and 0.1. As shown in Fig. 7, when the estimated channel information is imperfect, the MSE performances of GMP and SA-GMP with $L = 10$ are poor. When $L = 20$, the MSE of GMP becomes diverging severely and that of SA-GMP is improved slightly. Compared with the GMP and the SA-GMP, our proposed DNN-GMP and DNN-SA-GMP with $L = 10$ can still achieve reliable MSE performances at 7×10^{-4}.

5.5. *Impact without priori information*

To investigate the robustness of the proposed DNN-GMP and DNN-SA-GMP algorithms, we consider the signal recovery in the cases without the priori information, in which the variance vector of $\boldsymbol{x}$ is unavailable. Fig. 8 shows that the MSE performances of the GMP [18,19], the SA-GMP [18,19], the DNN-GMP, and the DNN-SA-GMP in the case of unavailable a priori variance.

Note that the initial point of MSE curve of the GMP at SNR= 20 dB is larger than 1 and MSE performance of the GMP becomes poor, which shows the convergence of the GMP diverges in the cases without a priori variance. For the SA-GMP without a priori



(a) Imperfect channel estimation with error variance 0.04

(b) Imperfect channel estimation with error variance 0.1

Fig. 7. MSEs comparisons of the proposed DNN-GMP, proposed DNN-SA-GMP, GMP [18,19], and SA-GMP [18,19] under imperfect channel estimation with error variances $\{0.1, 0.04\}$, where $K = 10$, $M = 20$, and $L \in \{10, 20\}$.

variance, its MSE performance become slightly bad when $L = 10$. As L increases to 20, the SA-GMP without a priori variance can achieve MSE as low as 2×10^{-3} when SNR $=$ 60 dB. Note that the initial points of MSE curves of DNN-GMP and DNN-SA-GMP are slightly higher than those of SA-GMP with $L = 10$ and 20. The reason is that the DNN-GMP and DNN-SA-GMP are trained under SNR=60 dB and the MSE performances of DNN-GMP and DNN-SA-GMP become slightly worse at SNR=20 dB. In contrast, the DNN-GMP without a priori variance can achieve MSE at 9×10^{-4} and the SA-GMP without a priori variance can achieve MSE as low as 5×10^{-4} when SNR $=$ 60 dB and $L = 10$. This verifies that the proposed DNN-GMP and DNN-SA-GMP are robust to the cases without priori variance.

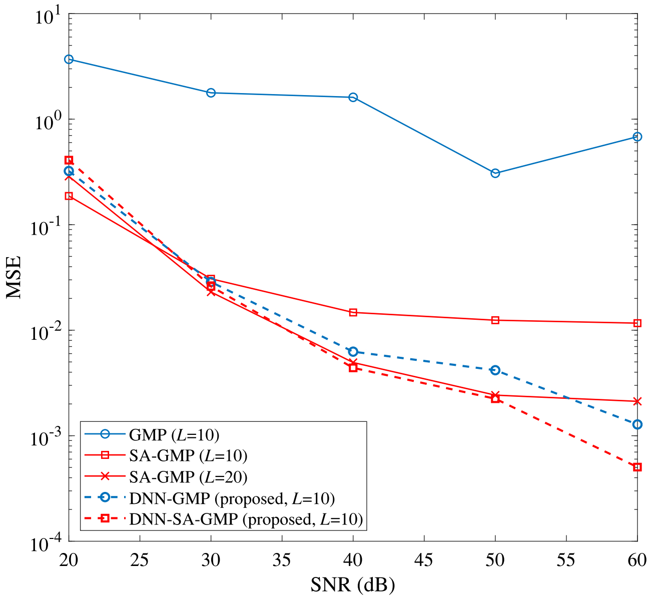


Fig. 8. MSEs comparisons of the proposed DNN-GMP, proposed DNN-SA-GMP, GMP [18,19], and SA-GMP [18,19] in the cases of unavailable priori variances.

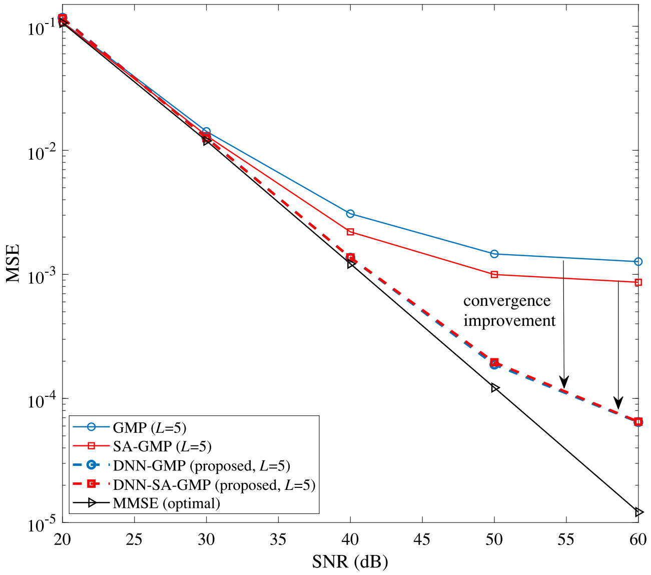


Fig. 9. MSE comparisons among the proposed DNN-GMP, the proposed DNN-SA-GMP, GMP [18,19], SA-GMP [18,19], and MMSE (optimal), where $K = 10$, $M = 60$, and $L = 5$.

5.6. *MSE performance in large-scale system*

To evaluate the effectivity of the proposed DNN-GMP and DNN-SA-GMP algorithms in large-scale systems, we consider the MSE performances of the GMP [18,19], the SA-GMP [18,19], the DNN-GMP, the DNN-SA-GMP, and MMSE over MIMO-NOMA system, where $K = 10$, $M = 60$, and $L = 5$. As shown in Fig. 9, the MSE performances of the DNN-GMP and the DNN-SA-GMP are more close to the MSE performance of MMSE than those of the GMP and the SA-GMP. Note that although the GMP and the SA-GMP can converge, the DNN-GMP and the DNN-SA-GMP improve the convergences significantly at the same iteration $L = 5$ with the aid of DNN. This verifies that the proposed DNN-GMP and DNN-SA-GMP algorithms can also be applicable to the large-scale systems.

6. Conclusion

In this paper, we have proposed the DNN-aided GMP algorithm and the DNN-aided SA-GMP algorithm to recover signals in URLLC systems, which combines with the advantages of DNN and MPA. With the aid of deep learning, training the constructed neural network offline is to search for the optimal weight and bias parameters. Numerical results have verified that the proposed DNN-aided GMP algorithm and DNN-aided SA-GAMP algorithm can achieve more reliable performance and faster convergence than the original GMP algorithm and SA-GAMP algorithm, as well as robust performances in the cases with imperfect channel estimations and the cases without a priori information.

There are three possible extensions to our work. The first one is to extend our proposed DNN-aided MPA methods in the discrete signals systems. The second one is to combine the DNN-aided MPA methods with sliding-window technology for very large-scale systems. The third extension is to exploit the evolutionary neural networks [48,49] to improve the adaptability of the proposed method in dynamic communication environments.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (No. 61772387, 61802296, 61750110529), China Postdoctoral Science Foundation Grant (No. 2017M620438), the Fundamental Research Funds for the Central Universities (JB180101), Fundamental Research Funds of Ministry of Education and China Mobile (MCM20170202), and also supported by the ISN State Key Laboratory.

References

[1] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, B. Shim, Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects, IEEE Wirel. Commun. 25 (3) (2018) 124–130.

[2] J. Guo, B. Song, Y. He, F.R. Yu, M. Sookhak, A survey on compressed sensing in vehicular infotainment systems, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2662–2680.

[3] G. Wu, S. Talwar, K. Johnsson, N. Himayat, K. Johnson, M2M: From mobile to embedded Internet, IEEE Commun. Mag. 49 (4) (2011) 36–43.

[4] B.P.L. Lau, N. Wijerathne, B.K.K. Ng, C. Yuen, Sensor fusion for public space utilization monitoring in a smart city, IEEE Internet Things J. 5 (2) (2018) 473–481.

[5] M. Bouaziz, A. Rachedi, A. Belghith, EKF-MRPL: Advanced mobility support routing protocol for internet of mobile things: Movement prediction approach, online published, https://doi.org/10.1016/j.future.2017.12.015.

[6] A.A. Khan, M.H. Rehmani, A. Rachedi, Cognitive-radio-based internet of things: Applications, architectures, spectrum related functionalities, and future research directions, IEEE Wirel. Commun. 24 (3) (2017) 17–25.

[7] Y.B. Zikria, M.K. Afzal, F. Ishmanov, S.W. Kim, H. Yu, A survey on routing protocols supported by the contiki internet of things operating system, Future Gener. Comput. Syst. 82 (2018) 200–219.

[8] J. Zuo, J. Zhang, C. Yuen, W. Jiang, W. Luo, Energy efficient user association for cloud radio access networks, IEEE Access 4 (2016) 2429–2438.

[9] F. Gabriel, A.K. Chorppath, I. Tsokalo, F.H. Fitzek, Multipath communication with finite sliding window network coding for ultra-reliability and low latency, arXiv preprint arXiv:1802.00521, 2018.

[10] T.V. Doan, G.T. Nguyen, A. Kropp, F.H. Fitzek, APMEC: An automated provisioning framework for multi-access edge computing, arXiv preprint arXiv:1805.09251, 2018.

[11] Y. Chi, L. Liu, G. Song, C. Yuen, Y.L. Guan, Y. Li, Message passing in c-ran: Joint user activity and signal detection, in: 2017 IEEE Global Communications Conference, 2017, pp. 1–6.

[12] F.R. Kschischang, B.J. Frey, H.A. Loeliger, Factor graphs and the sum-product algorithm, IEEE Trans. Inf. Theory 47 (2) (2001) 498–519.

[13] H.A. Loeliger, An introduction to factor graphs, IEEE Signal Proc. Mag. 21 (1) (2004) 28–41.

[14] D.L. Donoho, A. Maleki, A. Montanari, Message passing algorithms for compressed sensing, Proc. Natl. Acad. Sci. (2009).

[15] S. Rangan, Generalized approximate message passing for estimation with random linear mixing, arXiv preprint arXiv:1010.5141v2, 2010.

[16] L. Liu, C. Huang, Y. Chi, C. Yuen, Y.L. Guan, Y. Li, Sparse vector recovery: Bernoulli-Gaussian message passing, in: Proc. IEEE Global Communications Conference (GLOBECOM), 2017, pp. 1–6.

[17] T.J. Richardson, R.L. Urbanke, The capacity of low-density parity-check codes under message-passing decoding, IEEE Trans. Inf. Theory 47 (2) (2001) 599–618.

[18] L. Liu, C. Yuen, Y.L. Guan, Y. Li, Y. Su, A low-complexity Gaussian message passing iterative detector for massive MU-MIMO systems, in: Proc. International Conference on Information, Communications and Signal Processing (ICICS), 2015, pp. 1–5.

[19] L. Liu, C. Yuen, Y.L. Guan, Y. Li, Y. Su, Convergence analysis and assurance Gaussian message passing iterative detection for massive MU-MIMO systems, IEEE Trans. Wirel. Commun. 15 (9) (2016) 6487–6501.

[20] L. Liu, C. Yuen, Y.L. Guan, Y. Li, C. Huang, Gaussian message passing iterative detection for MIMO-NOMA systems with massive users, in: Proc. IEEE Global Communications Conference (GLOBECOM), 2016, pp. 1–6.

[21] C. Huang, L. Liu, C. Yuen, S. Sun, A LSE and sparse message passing-based channel estimation for mmWave MIMO systems, in: Proc. IEEE Globecom Workshops (GC Wkshps), 2016, pp. 1–6.

[22] Y. Chi, L. Liu, G. Song, C. Yuen, Y.L. Guan, Y. Li, Message passing in C-RAN: Joint user activity and signal detection, in: Proc. IEEE Global Communications Conference (GLOBECOM), 2017, pp. 1–6.

[23] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT press, 2016.

[24] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.C. Chen, L. Hanzo, Machine learning paradigms for next-generation wireless networks, IEEE Wirel. Commun. 24 (2) (2017) 98–105.

[25] T. Wang, C.K. Wen, H. Wang, F. Gao, T. Jiang, S. Jin, Deep learning for wireless physical layer: Opportunities and challenges, China Commun. 14 (11) (2017) 92–111.

[26] M. Chen, U. Challita, W. Saad, C. Yin, M. Debbah, Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks, arXiv preprint arXiv:1710.02913, 2017.

[27] N. Farsad, A. Goldsmith, Detection algorithms for communication systems using deep learning, arXiv preprint arXiv:1705.08044, 2017.

[28] T.J. O'Shea, T. Erpek, T.C. Clancy, Deep learning based MIMO communications, arXiv preprint arXiv:1707.07980, 2017.

[29] H. Ye, G.Y. Li, B.H. Juang, Power of deep learning for channel estimation and signal detection in OFDM systems, IEEE Wirel. Commun. Lett. 7 (1) (2018) 114–117.

[30] J.R. Hershey, J.L. Roux, F. Weninger, Deep unfolding: Model-based inspiration of novel deep architectures, arXiv preprint arXiv:1409.2574, 2014.

[31] E. Nachmani, E. Marciano, L. Lugosch, W.J. Gross, D. Burshtein, Y. Beery, Deep learning methods for improved decoding of linear codes, IEEE J. Sel. Top. Sign. Proces. 12 (1) (2018) 119–131.

[32] N. Samuel, T. Diskin, A. Wiesel, Deep MIMO detection, in: 2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2017, pp. 1–5.

[33] V. Kafedziski, Rate allocation for transmission of two gaussian sources over multiple access fading channels, IEEE Commun. Lett. 16 (11) (2012) 1784–1787.

[34] Y.I. Abramovich, N.K. Spencer, A.Y. Gorokhov, GLRT-based threshold detection-estimation performance improvement and application to uniform circular antenna arrays, IEEE Trans. Signal Process. 55 (1) (2007) 20–31.

[35] Y.I. Abramovich, N.K. Spencer, A.Y. Gorokhov, Detection-estimation of more uncorrelated gaussian sources than sensors in nonuniform linear antenna arrays - part iii: detection-estimation nonidentifiability, IEEE Trans. Signal Process. 51 (10) (2003) 2483–2494.

[36] X. Liu, O. Simeone, E. Erkip, Energy-efficient sensing and communication of parallel gaussian sources, IEEE Trans. Commun. 60 (12) (2012) 3826–3835.

[37] T.M. Cover, J.A. Thomas, Elements of Information Theory, second ed., Wiley, New York, NY, USA, 2006.

[38] A.E. Gamal, Y.-H. Kim, Elements of Information Theory, 2012.

[39] S. Gadkari, K. Rose, Time-division versus superposition coded modulation schemes for unequal error protection, IEEE Trans. Commun. 47 (3) (1999) 370–379.

[40] U. Wachsmann, R.F.H. Fischer, J.B. Huber, Multilevel codes: theoretical concepts and practical design rules, IEEE Trans. Inform. Theory 45 (5) (1999) 1361–1391.

[41] M. Borgerding, P. Schniter, S. Rangan, AMP-inspired deep networks for sparse linear inverse problems, IEEE Trans. Signal Process. 65 (16) (2017) 4293–4308.

[42] D. Tse, P. Viswanath, Fundamentals of Wireless Communication., Cambridge Univ. Press, Cambridge, U.K., 2005.

[43] J. Zhang, C.-K. Wen, C. Yuen, S. Jin, X. Gao, Large system analysis of cognitive radio network via partially-projected regularized zero-forcing precoding, IEEE Trans. Wirel. Commun. 14 (9) (2015) 4934–4947.

[44] L. Dai, B. Wang, Y. Yuan, S. Han, C.-L. I, Z. Wang, Non-orthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends, IEEE Commun. Mag. 53 (9) (2015) 74–81.
[45] S. Abeywickrama, L. Liu, Y. Chi, C. Yuen, Over-the-air implementation of uplink NOMA, in: Proc. IEEE GLOBECOM, 2017, pp. 1–6.
[46] Y. Chi, L. Liu, G. Song, C. Yuen, Y.L. Guan, Y. Li, Practical MIMO-NOMA: Low complexity & capacity-approaching solution, IEEE Trans. Wirel. Commun. 17 (9) (2018) 6251–6264.
[47] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in: Proc. of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2016, pp. 265–283.
[48] X. Yao, A review of evolutionary artificial neural networks, Int. J. Intell. Syst. 8 (4) (2010) 539–567.
[49] A. Baldominos, Y. Saez, P. Isasi, Evolutionary convolutional neural networks: An application to handwriting recognition, Neurocomputing 283 (29) (2018) 38–52.

Jie Guo received the B.S. degree in communication engineering from Zhengzhou University, Zhengzhou, China, in 2011. She received the Ph.D. degree from Xidian University, Xi'an, China, in 2017. From 2015 to 2016, she got the state scholarship fund from China scholarship council to be an exchange Ph.D. student with Carleton University, Canada. Her research interests include wireless communication, information fusion, deep learning and compressed video sensing.

Bin Song received his BS, MS, and PhD in communication and information systems from Xidian University, Xi'an, China in 1996, 1999, and 2002, respectively. He is currently a professor at the Xidian University, Xi'an, China. He has authored over 60 journal papers or conference papers and 40 patents. His research interests are in distributed video coding, compressed sensing-based video coding, content-based image recognition and machine learning, deep reinforcement learning, Internet of Things, big data.

Yuhao Chi received his B.S. degree in electronic and information engineering from Shaanxi University of Science Technology, Xi'an, China, in 2012, and Ph.D. degree in communication and information systems from Xidian University, Xi'an, China, in 2018. From 2016 to 2017, he got the state scholarship fund from China scholarship council to be an exchange Ph.D. student with Nanyang Technological University, Singapore, and a visiting student with the Singapore University of Technology and Design, Singapore. His research interests include coding theory, multiuser coding and detection, message passing algorithm, and deep learning.

Lahiru Jaysinghe received the BSc (Hons) degree from the Department of Electronic and Telecommunication Engineering, University of Moratuwa, Moratuwa, Sri Lanka, in 2016. He then worked in the industry for one year in the field of electronic design automation. Then he joined the Singapore University of Technology and Design, Singapore, as a Researcher. His current research interests include machine learning, deep learning, signal processing, scientific data mining, and pattern analysis methods for practical problems.

Chau Yuen received the BEng and PhD degree from Nanyang Technological University (NTU), Singapore, in 2000 and 2004 respectively. He is the recipient of Lee Kuan Yew Gold Medal, Institution of Electrical Engineers Book Prize, Institute of Engineering of Singapore Gold Medal, Merck Sharp Dohme Gold Medal and twice the recipient of Hewlett Packard Prize. Dr Yuen was a Post Doc Fellow in Lucent Technologies Bell Labs, Murray Hill during 2005. He was a Visiting Assistant Professor of Hong Kong Polytechnic University in 2008. During the period of 2006–2010, he worked at the Institute for Infocomm Research (I2R, Singapore) as a Senior Research Engineer, where he was involved in an industrial project on developing an 802.11n Wireless LAN system, and participated actively in 3Gpp Long Term Evolution (LTE) and LTE-Advanced (LTE-A) standardization. He joined the Singapore University of Technology and Design as an assistant professor from June 2010, and received IEEE Asia-Pacific Outstanding Young Researcher Award on 2012. Dr Yuen serves as an Associate Editor for IEEE Transactions on Vehicular Technology, and awarded as Top Associate Editor from 2009–2012.

Yong Liang Guan received his Ph.D. degree from the Imperial College of Science, Technology and Medicine, University of London, in 1997, and B.Eng. degree with first class honors from the National University of Singapore in 1991. He is now an associate professor at the School of Electrical and Electronic Engineering, Nanyang Technological University. His research interests include modulation, coding and signal processing for communication, information security and storage systems. The author's homepage is available online at http://www3.ntu.edu.sg/home/eylguan.

Xiaojiang (James) Du is a tenured professor in the Department of Computer and Information Sciences at Temple University, Philadelphia, USA. Dr. Du received his M.S. and Ph.D. degrees in electrical engineering from the University of Maryland College Park in 2002 and 2003, respectively. His research interests are wireless communications, wireless networks, security, and systems. He has authored over 300 journal and conference papers in these areas as well as a book, published by Springer. He won the best paper award at IEEE GLOBECOM 2014 and the best poster runner-up award at the ACM MobiHoc 2014. Dr. Du served as the lead Chair of the Communication and Information Security Symposium of the IEEE International Communication Conference (ICC) 2015 and a Co-Chair of Mobile and Wireless Networks Track of IEEE Wireless Communications and Networking Conference (WCNC) 2015. He is (was) a Technical Program Committee (TPC) member of several premier ACM/IEEE conferences. Dr. Du is a Senior Member of IEEE and a Life Member of ACM.

Mohsen Guizani (S'85–M'89–SM'99–F'09) received the B.S. (with distinction) and M.S. degrees in electrical engineering, the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor at the CSE Department in Qatar University, Qatar. Previously, he served in different academic and administrative positions at the University of Idaho, Western Michigan University, University of West Florida, University of Missouri-Kansas City, University of Colorado-Boulder, and Syracuse University. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He is currently the Editor-in-Chief of the IEEE Network Magazine, serves on the editorial boards of several international technical journals and the Founder and Editor-in-Chief of Wireless Communications and Mobile Computing journal (Wiley). He is the author of nine books and more than 500 publications in refereed journals and conferences. He guest edited a number of special issues in IEEE journals and magazines. He also served as a member, Chair, and General Chair of a number of international conferences. Throughout his career, he received three teaching awards and four research awards. He also received the 2017 IEEE Communications Society WTC Recognition Award as well as the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer. He is a Fellow of IEEE and a Senior Member of ACM.