PAPER Special Section on Information Theory and Its Applications

User Identification and Channel Estimation by Iterative DNN-Based Decoder on Multiple-Access Fading Channel

Lantian WEI[†a)], ***Student Member***, **Shan LU**[††b)], ***Member***, **Hiroshi KAMABE**[††c)], ***Senior Member***, ***and*** **Jun CHENG**[†††d)], ***Member***

SUMMARY In the user identification (UI) scheme for a multiple-access fading channel based on a randomly generated $(0, 1, -1)$-signature code, previous studies used the signature code over a noisy multiple-access adder channel, and only the user state information (USI) was decoded by the signature decoder. However, by considering the communication model as a compressed sensing process, it is possible to estimate the channel coefficients while identifying users. In this study, to improve the efficiency of the decoding process, we propose an iterative deep neural network (DNN)-based decoder. Simulation results show that for the randomly generated $(0, 1, -1)$-signature code, the proposed DNN-based decoder requires less computing time than the classical signal recovery algorithm used in compressed sensing while achieving higher UI and channel estimation (CE) accuracies.

key words: *signature code, deep neural network, compressed sensing, user identification, channel estimation*

1. Introduction

In a wireless network with a large number of devices, such as massive machine-type communication (mMTC), most of the time, only a fraction of users are simultaneously active. In this context, user identification (UI) for a multiple-access fading channel is an important issue. The UI scheme based on signature code is widely concerned because the code length L is smaller than the number of devices T, which leads to a higher spectral efficiency than other conventional multiple-access-based schemes. In previous studies, the signature code was constructed for a noisy multiple-access adder channel, and only the user state information (USI) was decoded using the signature decoder [1]–[3]. However, in a multi-access fading channel, the channel state information (CSI) is also important for channel decoding.

Consider a signature matrix $\boldsymbol{S}$ and a fading coefficient vector $\boldsymbol{k}$ with elements given by the multiplication of the active status of user and channel coefficients. Because only a fraction of users are simultaneously active, $\boldsymbol{k}$ is sparse. The recovery of the vector $\boldsymbol{k}$ from the superimposed signal $\boldsymbol{y} = \boldsymbol{k}\boldsymbol{S} + \boldsymbol{n}$ with the highest possible accuracy is a basic problem in compressed sensing. The recovered vector contains not only the USI but also the CSI. This idea makes the joint channel estimation (CE) and UI possible.

In compressed sensing field, the ℓ_1-regularized least squares (LS) algorithm that minimizes $\|\boldsymbol{y} - \boldsymbol{k}\boldsymbol{S}\|_2^2 + \lambda \|\boldsymbol{k}\|_1$ is commonly used for solving sparse signal recovery problems. There are several algorithms to solve ℓ_1-regularized LS problems efficiently, among which the least absolute shrinkage and selection operator (LASSO) is well known. The iterative shrinkage thresholding algorithm (ISTA) [4] is one of the most well-known algorithms for solving LASSO problems; it is an iterative algorithm comprising a linear estimation and a shrinkage process based on a soft thresholding function. An interior-point method was also proposed for solving large-scale ℓ_1-regularized least-squares programs (LSPs) to compute the search direction in a sparse reconstruction algorithm based on the preconditioned conjugate gradient algorithm [5]. The matching pursuit (MP) algorithm and orthogonal matching pursuit (OMP) [6], [7] are also ℓ_1-regularization algorithms with incomplete measurements. Above iterative sparse reconstruction algorithms are usually used as iterative joint CE and UI solutions in previous studies with similar settings, like [8], [9].

Machine learning (ML) algorithms enable one to perform tasks that are too challenging to perform with fixed programs written and designed by humans [10]. Standard deep-learning techniques, such as stochastic gradient descent (SGD) algorithms based on mini-batches, are used to adjust the trainable variables for improving the algorithm performance. Recently, ML, including deep neural networks (DNNs), has been applied as a promising solution for practical applications, such as a stable and efficient decoder [11], [12]. In the field of signal processing, algorithms based on ML have been applied to recover sparse signals [13], [14]. In related research on signature codes, by applying training data to a properly designed DNN, the scheme proposed in [15] learns the nonlinear mapping between the received signal and the support for detecting active users. Unlike traditional algorithms and ML methods based on conventional models, neural network-based solutions can improve the model's performance through a large amount of data, which is not present in traditional ML because of the

Manuscript received February 19, 2021.
Manuscript revised June 18, 2021.
Manuscript publicized September 1, 2021.

[†]The author is with the Graduate School of Engineering, Gifu University, Gifu-shi, 501-1193 Japan.
[††]The authors are with the Department of Electrical, Electronic and Computer Engineering, Gifu University, Gifu-shi, 501-1193 Japan.
[†††]The author is with the Department of Intelligent Information Engineering and Sciences, Doshisha University, Kyotanabe-shi, 610-0321 Japan.
a) E-mail: z3921004@edu.gifu-u.ac.jp
b) E-mail: slu@gifu-u.ac.jp
c) E-mail: kamabe@m.ieice.org
d) E-mail: jcheng@ieee.org
DOI: 10.1587/transfun.2021TAP0008

influence of the standard model used.

In our previous study [16], a DNN-based decoder was implement with a binary signature code proposed in [3]. However, that signature code was constructed for a noisy multiple-access adder channel, not perform well in a fading channel. Furthermore, the preprocessing of the decoder is too complicated. In the present study, we consider a randomly generated $(0, 1, -1)$-signature code for a multi-access wireless fading channel and propose an iterative DNN-based decoder for the code to identify active users and estimate the channel coefficients for active users. The DNN-based decoder contains two types of basic units: a DNN-based user detector and a DNN-based channel estimator. A recursive algorithm is designed between the DNN-based detector and DNN-based estimator. Because the DNN-based detector provides prior information to the DNN-based channel estimator, the DNN-based estimator provides highly accurate estimation. Furthermore, simulation results show that the proposed DNN-based decoder achieves higher active user-detection and channel-estimation accuracies than existing algorithms derived from compressed sensing technology for the $(0, 1, -1)$-signature code.

2. System Model

We consider a multiple-access fading communication system in which T users communicate with a base station, as shown in Fig. 1. A small fraction of these users are active simultaneously. Let $x_i \in \{0, 1\}$ represent the active status of the i-th user, where 0 represents idle and 1 represents active, and let $P(x_i = 1)$ denote the probability of user activity. We assume that the active statuses of these users are independent and that the probability of being active is uniform and equal to ρ for all users, that is, $P(x_i = 1) = \rho$ for any i.

To determine the active status of the base station, when the i-th user is active, a unique codeword $\boldsymbol{s}_i \in \{0, 1, -1\}^L$ (code length $L < T$) is sent. The base station receives a superimposed signal $\boldsymbol{y}$ expressed as

$$\boldsymbol{y} = \sum_{i=1}^{T} x_i h_i \boldsymbol{s}_i + \boldsymbol{n}, \tag{1}$$

where h_i is the channel coefficient for the i-th user, which is a random variable following the Rayleigh distribution, and

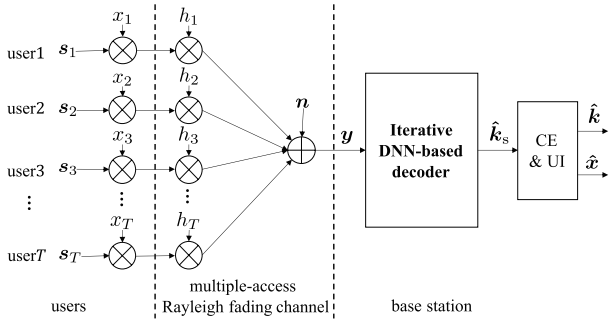

Fig. 1 T-user multiple-access communication over a wireless fading channel.

$\boldsymbol{n} \in \mathbb{R}^L$ is a Gaussian noise vector.

Let $\boldsymbol{k} = (k_1, k_2, \ldots, k_T)$ with $k_i = x_i h_i$, and let $\mathcal{S}$ be a $T \times L$ matrix $\mathcal{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_T]^{\mathrm{T}}$. We simplify Eq. (1) as follows:

$$\begin{aligned} \boldsymbol{y} &= \sum_{i=1}^{T} k_i \boldsymbol{s}_i + \boldsymbol{n} \\ &= \boldsymbol{k}\mathcal{S} + \boldsymbol{n}. \end{aligned} \tag{2}$$

Note that $\boldsymbol{k} = (k_1, k_2, \ldots, k_T)$ is the fading coefficient vector for active users. The compression ratio of the signature matrix is defined as T/L and usually set to 2.

At the receiver, from the received signal $\boldsymbol{y}$, we estimate the fading coefficient vector $\boldsymbol{k}$ by solving the following problem:

$$\hat{\boldsymbol{k}} = \underset{\boldsymbol{k} \in (\mathbb{R}_{\geq 0})^T}{\arg\min} \|\boldsymbol{y} - \boldsymbol{k}\mathcal{S}\|_2^2, \ \text{s.t. } P(k_i \neq 0) = \rho, \tag{3}$$

where $\|\cdot\|_p$ denotes the ℓ_p norm. For a real number $p \geq 1$, the ℓ_p norm of the vector $\boldsymbol{v}$ is defined by

$$\|\boldsymbol{v}\|_p = (|v_1|^p + |v_2|^p + \cdots + |v_n|^p)^{1/p}, \tag{4}$$

and when $p = 0$, it is the number of non-zero elements in a vector; for instance, $\|(0, 3, 2, 0)\|_0 = 2$.

In this paper, we first propose a DNN-based decoder to provide soft information about $\boldsymbol{k}$, $\hat{\boldsymbol{k}}_{\mathrm{s}} = (\hat{k}_{\mathrm{s},1}, \hat{k}_{\mathrm{s},2}, \ldots, \hat{k}_{\mathrm{s},T})$. Next, from $\hat{\boldsymbol{k}}_{\mathrm{s}}$, the user status $\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_T)$ is estimated with a positive threshold τ as

$$\hat{x}_i = \begin{cases} 0 & \hat{k}_{\mathrm{s},i} < \tau \\ 1 & \hat{k}_{\mathrm{s},i} \geq \tau. \end{cases} \tag{5}$$

Finally, we obtain an accurate estimate of $\boldsymbol{k}$:

$$\hat{\boldsymbol{k}} = (\hat{x}_1 \hat{k}_{\mathrm{s},1}, \hat{x}_2 \hat{k}_{\mathrm{s},2}, \ldots, \hat{x}_T \hat{k}_{\mathrm{s},T}). \tag{6}$$

3. Feedforward Neural Network

Before introducing our DNN-based decoder, we first review the feedforward neural network (FNN) structure that will be used later.

The FNN is one of the most commonly used neural networks. A basic FNN structure is composed of one input layer, N repetitive hidden layers, and one output layer. Each hidden layer of the basic FNN unit is composed of a weight matrix, W_i; a bias vector, $\boldsymbol{b}_i$; and an activation function, ϕ_i, where i denotes the index of the hidden layer. The output layer is composed of a weight matrix, W_{o}; a bias vector, $\boldsymbol{b}_{\mathrm{o}}$; and an activation function, ϕ_o. The output of the FNN with N hidden layers is expressed as

$$\begin{aligned} &f(\boldsymbol{v}; \Theta) \\ &\quad = \phi_o(W_o \phi_N(W_N \phi_{N-1}(\ldots \phi_1(W_1 \boldsymbol{v} \\ &\qquad + \boldsymbol{b}_1) \ldots) + \boldsymbol{b}_N) + \boldsymbol{b}_o), \end{aligned} \tag{7}$$

where $\boldsymbol{v}$ denotes the input of the FNN unit and Θ is the

Table 1 Parameters of the FNN models.

Model	Number of hidden layers	Hidden layer's neural
Simple model	1	$T + L$
Middle model	3	$T + L$
Wide model	3	$(T + L) \times 2$
Deep model 1	6	$T + L$
Deep model 2	14	$T + L$

set of weight matrices and bias vectors for FNN, that is, $\Theta = \{W_i, \boldsymbol{b}_i | i = 0, 1, \ldots, N\} \cup \{W_o, \boldsymbol{b}_o\}$.

Two types of activation functions are used in this study. The first is the rectified linear unit (ReLU) [17], which is expressed as

$$\phi_R(x) = x_+ = \max(0, x). \tag{8}$$

ReLU is a non-linear function with the domain $(-\infty, +\infty)$ and range $[0, +\infty)$. The second activation function is the sigmoid function. The standard sigmoid function is expressed as follows:

$$\phi_s(x) = \frac{1}{1 + e^{-x}}. \tag{9}$$

It is a nonlinear function with the domain $(-\infty, +\infty)$ and range $(0, 1)$.

4. Decoder Model Based on Simple FNN Structure

In this section, we built a decoder based on a simple FNN structure called a simple FNN-based decoder. We tried five FNN models with different parameters and gave the training results.

4.1 Parameters of FNN Models

By adjusting the number of layers and number of nodes of the FNN, a simple model, a middle model, a wide model, and two deep models were created. These FNN models use ReLU as the activation function in the hidden and output layers, the Table 1 shows the structure details.

4.2 Loss Function for the Simple FNN-Based Decoder

The decoder recovers the fading coefficient vector $\boldsymbol{k}$ by receiving the signal $\boldsymbol{y}$. Therefore, the input of the FNNs is $\boldsymbol{y}$, and we expect the output of FNNs $f(\boldsymbol{y}; \Theta)$ to be close to the fading coefficient vector $\boldsymbol{k}$. To represent the difference between $f(\boldsymbol{y}; \Theta)$ and $\boldsymbol{k}$, we use the mean squared error (MSE):

$$\sum_{i=1}^{T} (k_i - f(\boldsymbol{y}; \Theta)_i)^2 / T, \tag{10}$$

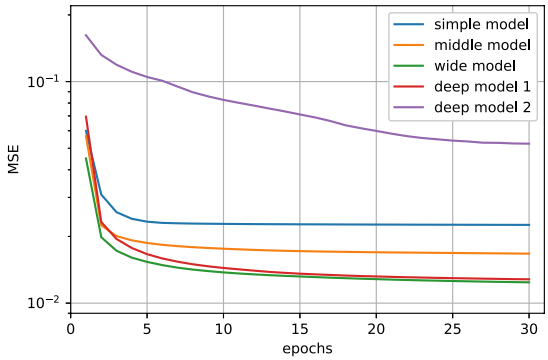

Fig. 2 Training curve of the simple FNN-based decoder.

which is also used as a loss function in training.

4.3 Training Results of the Simple FNN-Based Decoder

We trained each decoder using training data which randomly generated by a fixed $(0, 1, -1)$-signature matrix with size 100×50, when setting $\rho = 0.1$, and SNR= 10 dB. The training curves shown in Fig. 2. In the training the wide model and deep model 1 show the best performance, the middle model shows the next-best performance, and deep model 2 shows the worst performance.

In the simulation, increasing the number of hidden layers and the number of nodes in a hidden layer within a certain range can improve the performance of the model. Although increasing the number of nodes in the hidden layer can increase the performance, the network complexity increases exponentially as the number of nodes in a hidden layer increases. Another technique to improve performance is to increase the number of hidden layers. However, if the number of hidden layers is too large, the training of the neural network becomes difficult, and the model performance begins to decrease. Therefore, the performance improvement cannot rely solely on the simple stacking of neural networks, we need to design the network further. In addition, the number of the trainable parameters of the wide model is 3.6 times to the middle model and 1.46 times to the deep model 1. To obtain the trade-off between complexity and performance to the FNN, we choose $L + T$ as the number of nodes in each hidden layer in the following proposed DNN-based decoder.

5. Proposed Method

In this section, we first describe how the superimposed signal $\boldsymbol{y}$ is recovered from the fading coefficient vector $\boldsymbol{k}$ in our proposed DNN-based decoder. Subsequently, we present a training procedure for the proposed scheme.

5.1 DNN-Based Decoder Structure

To solve the optimization problem (3), we propose an iterative DNN-based decoder consisting of several generations, as shown in Fig. 3. In the m-th generation of the DNN-based decoder, the soft information on the user status, denoted by

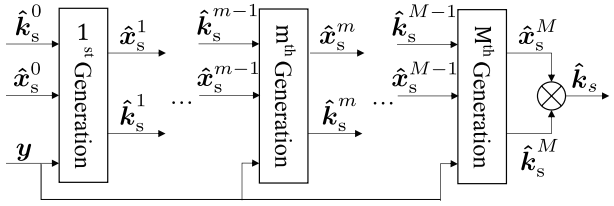

Fig. 3 Iterative DNN-based decoder with M generations.

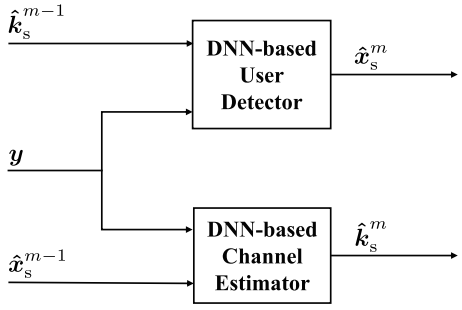

Fig. 4 Schematic of the m-th generation of the DNN-based decoder.

$\hat{\boldsymbol{x}}_s^m$, and the estimate of the fading coefficient vector $\boldsymbol{k}$ of active users, denoted by $\hat{\boldsymbol{k}}_{\mathrm{s}}^m$, are obtained, given the received signal $\boldsymbol{y}$ and the previous generation's outputs $\hat{\boldsymbol{x}}_s^{m-1}$ and $\hat{\boldsymbol{k}}_{\mathrm{s}}^{m-1}$. For the first generation, the input $\hat{\boldsymbol{x}}_s^0$ is defined as an all-zero vector $(0,0,\ldots,0) \in \mathbb{R}^T$, and $\hat{\boldsymbol{k}}_{\mathrm{s}}^0$ is defined as an all-one vector $(1,1,\ldots,1) \in \mathbb{R}^T$.

As shown in Fig. 4, the m-th generation of the DNN-based decoder contains a user detector and a channel estimator, named DNN-based user detector and DNN-based channel estimator, respectively.

Given the received signal $\boldsymbol{y}$ and the previous estimator's output $\hat{\boldsymbol{k}}_{\mathrm{s}}^{m-1}$, the DNN-based user detector estimates the soft information on the user status as

$$\hat{\boldsymbol{x}}_{\mathrm{s}}^m = f_{\mathrm{d}}((\boldsymbol{y}, \hat{\boldsymbol{k}}_{\mathrm{s}}^{m-1}); \Theta_{\mathrm{d}}), \tag{11}$$

where f_d is the output function of the FNN for the DNN-based channel estimator (7) and Θ_{d} denotes the set of weights and biases of the DNN-based user detector. The DNN-based channel estimator provides the received signal $\boldsymbol{y}$ and the previous detector's output $\hat{\boldsymbol{x}}_{\mathrm{s}}^{m-1}$ as

$$\hat{\boldsymbol{k}}_{\mathrm{s}}^m = f_{\mathrm{e}}((\boldsymbol{y}, \hat{\boldsymbol{x}}_{\mathrm{s}}^{m-1}); \Theta_{\mathrm{e}}), \tag{12}$$

where f_e is the output function of the FNN for the DNN-based channel estimator (7) and Θ_{e} denotes the set of weights and biases of the DNN-based estimator. Note that the DNN-based channel estimator aims to solve the problem of minimizing $\|\boldsymbol{y} - \boldsymbol{k}\mathcal{S}\|_2^2$. This problem always has a solution, but the solution may not be unique. Therefore, the DNN-based user detector is necessary to provide prior information on the position of the non-zero elements in the vector $\boldsymbol{k}$. The prior information makes the solution of the minimization problem unique or approximately unique. The DNN-based user detector can also be viewed as a shrinkage process for channel estimation.

Table 2 Parameters of the neural network.

Parameter	Estimator	Detector
Number of hidden layers	3	3
Input layer's neural	$T+L$	$T+L$
Hidden layer's neural	$T+L$	$T+L$
Output layer's neural	T	T

Table 3 Activation functions.

Layer	Estimator	Detector
Main hidden layer	ReLU	ReLU
Last hidden layer	ReLU	None
Output layer	None	Sigmoid

The DNN-based user detector and DNN-based channel estimator both use the FNN structure described in Sect. 3. The parameters and activation functions of the FNNs are listed in Tables 2 and 3, respectively. In addition, we add a batch-normalization layer before each hidden layer [18] for substantially accelerating the training of deep networks. The batch-normalization layer reparametrizes the underlying optimization problem to make it stabler and smoother. Consequently, the neural network can converge faster during training [19].

5.2 Training Procedure

The proposed FNN structure is trained with the detector, $f_{\mathrm{d}}((\boldsymbol{y}, \hat{\boldsymbol{k}}_{\mathrm{s}}^{m-1}); \Theta_{\mathrm{d}})$, and the estimator, $f_{\mathrm{e}}((\boldsymbol{y}, \hat{\boldsymbol{x}}_{\mathrm{s}}^{m-1}); \Theta_{\mathrm{e}})$, which can recover the fading coefficient vector $\boldsymbol{k}$. In the training procedure, a multi-loss function inspired by [11] is used:

$$\begin{aligned}&\mathrm{Loss}(\boldsymbol{y}, \hat{\boldsymbol{k}}_{\mathrm{s}}^0, \hat{\boldsymbol{x}}_{\mathrm{s}}^0, \ldots, \hat{\boldsymbol{k}}_{\mathrm{s}}^{M-1}, \hat{\boldsymbol{x}}_{\mathrm{s}}^{M-1}, \boldsymbol{k}, \boldsymbol{x}; \Theta_{\mathrm{d}}, \Theta_{\mathrm{e}}) \\ &\quad = \sum_{m=1}^{M} (\mathrm{MSE}_{\mathrm{d}}^m(\boldsymbol{y}, \hat{\boldsymbol{k}}_{\mathrm{s}}^{m-1}, \boldsymbol{x}; \Theta_{\mathrm{d}}) \\ &\qquad + \mathrm{MSE}_{\mathrm{e}}^m(\boldsymbol{y}, \hat{\boldsymbol{x}}_{\mathrm{s}}^{m-1}, \boldsymbol{k}; \Theta_{\mathrm{e}})),\end{aligned} \tag{13}$$

where $\mathrm{MSE}_{\mathrm{d}}^m$ denotes the MSE of the DNN-based user detector in m-th generation, which is defined as

$$\begin{aligned}&\mathrm{MSE}_{\mathrm{d}}^m(\boldsymbol{y}, \hat{\boldsymbol{k}}_{\mathrm{s}}^{m-1}, \boldsymbol{x}; \Theta_{\mathrm{d}}) \\ &\quad \triangleq \sum_{i=1}^{T} (f_{\mathrm{d}}((\boldsymbol{y}, \hat{\boldsymbol{k}}_{\mathrm{s}}^{m-1}); \Theta_{\mathrm{d}})_i - x_i)/T.\end{aligned} \tag{14}$$

$\mathrm{MSE}_{\mathrm{e}}^m$ denotes the MSE of the DNN-based estimator in the i-th generation, which is defined as

$$\mathrm{MSE}_{\mathrm{e}}^m(\boldsymbol{y}, \hat{\boldsymbol{x}}_{\mathrm{s}}^{m-1}, \boldsymbol{k}; \Theta_{\mathrm{e}})$$

$$\triangleq \sum_{i=1}^{T} (f_{\mathrm{e}}((\boldsymbol{y}, \hat{\boldsymbol{x}}_{\mathrm{s}}^{m-1}); \Theta_{\mathrm{e}})_i - k_i)/T. \tag{15}$$

Let $\theta_i \in \{\Theta_{\mathrm{d}} \cup \Theta_{\mathrm{e}}\}$ be a trainable parameter of the DNN-based decoder. The updated θ_i, denoted by θ_i^+, is obtained using the SGD method as follows:

$$\theta_i^+ := \theta_i - \alpha \frac{\partial}{\partial \theta_i} \mathrm{Loss}(\boldsymbol{y}, \ldots, \boldsymbol{k}, \boldsymbol{x}; \Theta_{\mathrm{d}}, \Theta_{\mathrm{e}}), \tag{16}$$

where α is the learning rate (step size). Note that other optimization algorithms based on the SGD method can also be used, such as Adam [20] and RMSprop [21].

6. Numerical Experiments

In this section, we present the DNN-based decoder's performance for various signal-to-noise ratios (SNRs). In the experiments, the $(0, 1, -1)$-signature matrix with size 100×50 was used, and the zero elements in the matrix were set according to a Bernoulli distribution with a probability of 0.5. Furthermore, non-zero elements take the values 1 or -1 according to the Bernoulli distribution with a probability of 0.5. The user activity probabilities ρ is fixed at 0.1, and the channel coefficient obeys the Rayleigh distribution, with the scale parameter being 1.

To make the neural network converge fully, we randomly generated 75,000 batches of training data to train the DNN-based decoder; the batch size was 1,000. The Xavier method was used to initialize the weights and biases [22]. The learning rate α is fixed at 1×10^{-3}. Figure 5 shows the training curve for the DNN-based decoder with five generations when SNR= 10 dB. Our trained batches are sufficient to stabilize the decoder's multi-loss. Figure 6 and Fig. 7 show the MSE of the detector and estimator, respectively, for different generations in the DNN-based decoder. The MSE tends to stabilize after three to five generations. Therefore, in the subsequent experiments, the number of generations in training and simulation were set to five.

6.1 User Identification Accuracy

Here, we present the performance of the DNN-based decoder in UI and CE. The status judgment error rate (SER) is defined as follows:

$$\mathrm{SER} \triangleq \mathbb{E}\left[\frac{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_0}{T}\right]. \tag{17}$$

Figure 8 shows the SER performance of the DNN-based decoder in various values of τ when SNR= 10 dB. The DNN-based decoder has the best performance when τ is in the interval $(0.01, 0.1)$, so we take an intermediate value 0.05 as the value of τ. Figure 9 shows the SER results in various generations of the DNN-based decoder for randomly generated $(0, 1, -1)$-signature matrices when τ was set to 0.05. The horizontal axis represents the SNR of this system. For the DNN-based decoder, the training SNR is equal to the testing

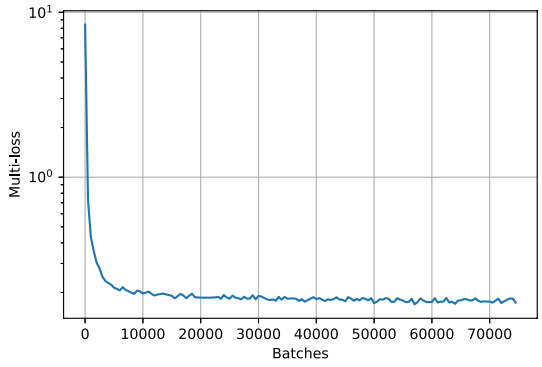

Fig. 5 Training curve of the DNN-based decoder, when SNR= 10 dB.

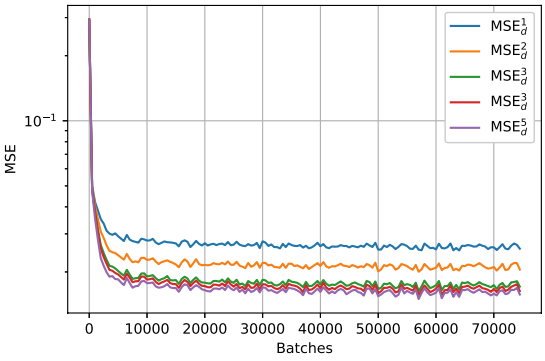

Fig. 6 MSE of the DNN-based detector for different generations in training, when SNR= 10 dB.

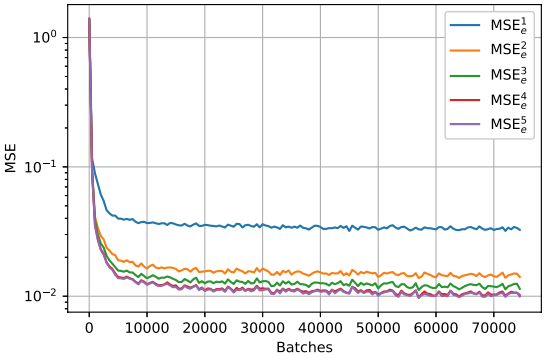

Fig. 7 MSE of the DNN-based estimator for different generations in training, when SNR= 10 dB.

SNR. The results in Fig. 9 show that as the number of iterations increases, the performance of the DNN-based decoder is better, which proves the validity of the iterative approach for the proposed DNN-based decoder. For comparison, the figure also shows the SER results for several classical recovery algorithms: ISTA [4]; OMP [7]; basis pursuit (BP) [23], which was implemented by disciplined convex programming [24] in our simulations; and large-scale ℓ_1-regularized LSPs [5]. The results show that the proposed DNN-based decoder achieves the lowest SER among the compared methods.

6.2 Channel Estimation Accuracy

To evaluate the CE accuracy of the decoders, we used the

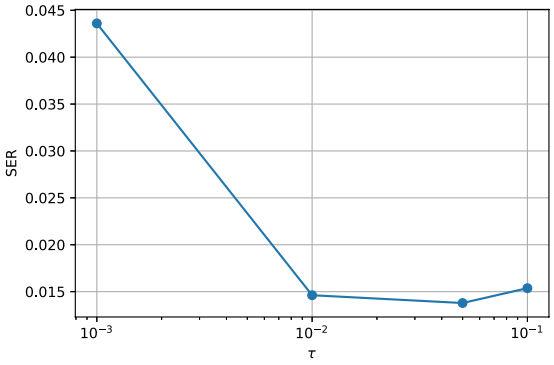

Fig. 8 SER performance of DNN-based decoder in various τ, when SNR= 10 dB.

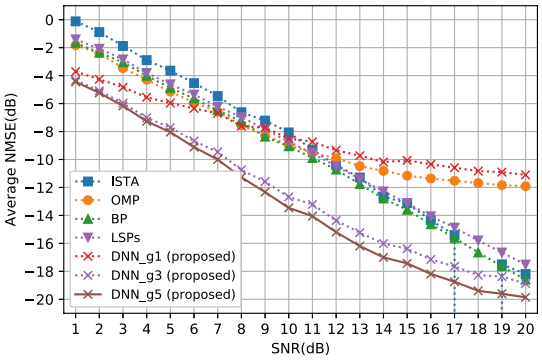

Fig. 10 NMSE performance of decoding with randomly generated $(0, 1, -1)$-signature code.

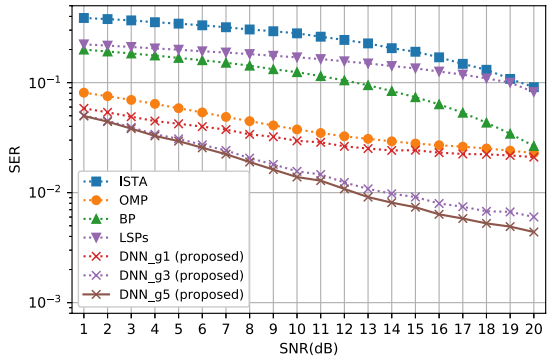

Fig. 9 SER performance of decoding with randomly generated $(0, 1, -1)$-signature code.

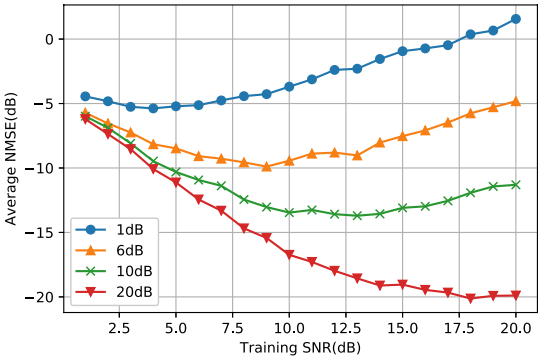

Fig. 11 NMSE performance of DNN-based decoder with randomly generated $(0, 1, -1)$-signature code.

average normalized MSE (NMSE), which is defined as follows:

$$\text{NMSE(dB)} \triangleq 10\log_{10}\mathbb{E}\left[\frac{\|\hat{\boldsymbol{k}}-\boldsymbol{k}\|_2^2}{\|\boldsymbol{k}\|_2^2}\right]. \tag{18}$$

The average NMSE reflects the gap between the estimated fading coefficient vector $\hat{\boldsymbol{k}}$ and the actual fading coefficient vector $\boldsymbol{k}$. A smaller average NMSE results in a better channel estimate accuracy of the decoder. To avoid a situation where the denominator is zero when calculating the NMSE, we excluded data corresponding to inactive users, that is, $\boldsymbol{x} \neq (0)^T$. Figure 10 shows the average NMSE performance of each decoder for a randomly generated $(0, 1, -1)$-signature matrix with various SNRs. For the DNN-based decoder, the training SNR is equal to the testing SNR. The results in Fig. 10 show that the validity of the iterative approach for the proposed DNN-based decoder, and the proposed DNN-based decoder achieves the lowest average NMSE among all the compared methods. For ISTA and BP, the NMSE is relatively high because, in the recursion of optimization, no prior information on the user status is provided. OMP is an iterative greedy algorithm that easily falls into a local optimal solution.

6.3 Generalization Ability Discussion

In order to test the generalization ability of the DNN decoder under different testing SNR, we conducted experiments on the decoder trained under different training SNR and perform these decoders at testing SNR of 1 dB, 6 dB, 10 dB, and 20 dB. The experimental results are shown in Fig. 11. It can be seen from Fig. 11 that the training SNR basically determines the performance limit of the decoder. The larger the training SNR be using, the better the performance of the decoder in the high testing SNR field, however when the training SNR greater than 8 dB, the performance of the decoder in the low testing SNR field becomes to accelerate deterioration, so we think testing SNR in the interval $(8, 13)$ is a more appropriate parameter. Since the data in training is completely set by the program and randomly generated, these parameters may deviate from the actual situation. We recommend using the data obtained in the working environment for neural network training, and this will be the best choice.

We also observed the performance of the decoder when the user's active probability ρ changes in testing when training $\rho = 0.1$, SNR= 10 dB, and the results are shown in Fig. 12. When the testing ρ is greater than the training ρ, the decoding difficulty increases due to the increase of active users in the same time slot, and the performance of the decoder will decrease. When the testing ρ is slightly smaller

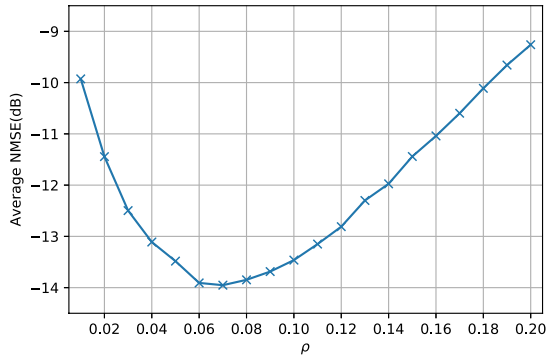

Fig. 12 NMSE performance of DNN-based decoder in various ρ, when SNR= 10 dB.

Table 4 Average computing time for various decoders (in seconds).

OMP	DNN	ISTA	LSPs	BP
4.15E-4	3.12E-3	4.09E-2	0.177	0.507

than the training ρ, the decoding difficulty is reduced due to the reduction of active users in the same time slot, and the performance of the decoder will be improved, like in the interval (0.07, 0.1). However, when ρ continues to decrease, although the decoding difficulty decreases, the received signal at this time is too different from the signal in the training phase, the performance of the decoder begins to deteriorate, like in the interval (0.01, 0.07). Therefore, the DNN-based decoder has a certain generalization ability for ρ, when ρ does not change much, like an interval (0.04, 0.12), its performance does not deteriorate.

6.4 Computation Efficiency

Table 4 lists the average computing times in the above experiment. The decoding methods explained in the previous subsections were performed off-line using a computer equipped with Intel Core i5 CPU at 3.10 GHz with 8 GB memory. To ensure fairness, no GPU was used in this test. The table demonstrates that the proposed DNN-based decoder is less time consuming than those based on BP, ISTA, and LSPs for recovering the fading coefficient vector $\boldsymbol{k}$. However, the computing time of the OMP-based decoder is less than that of the DNN-based decoder. This is because the OMP-based decoder is a greedy algorithm that considers only the local minimum solution and not the global solution. Consequently, the performance of the OMP-based decoder is significantly lower than that of the DNN-based decoder in user detection as well as channel estimation. Therefore, the DNN-based decoder is the most efficient decoder.

7. Conclusions

In this study, we considered a randomly generated $(0, 1, -1)$-signature code for a multi-access wireless fading channel, in which a small fraction of users are active simultaneously, and proposed an iterative DNN-based decoder for the code to identify active users and estimate the channel coefficients. A recursive algorithm was designed between the DNN-based detector and DNN-based estimator. Because the DNN-based detector provides prior information to the DNN-based channel estimator, the DNN-based estimator achieves highly accurate estimation.

Simulation results show that the proposed DNN-based decoder achieves higher accuracies in UI and CE than existing algorithms derived from compressed sensing technology for the randomly generated $(0, 1, -1)$-signature code.

As the performance of sparse signal recovery in compressed sensing depends substantially on the compression matrix used, in the future, we will investigate the design of a signature matrix that can achieve good sparse signal recovery performance in a wireless fading channel.

Acknowledgments

We would like to thank T. Wadayama for his useful comments. This work was supported in part by the Japan Society for the Promotion of Science for Scientific Research (C) under Grant 18K04132 and Grant 20K04493 and in part by Grants-in-aid for Promotion of Regional Industry-University-Government Collaboration from the Cabinet Office, Japan.

References

[1] J. Cheng, K. Kamoi, and Y. Watanabe, "Error-correcting signature code for multiple-access adder channel," Proc. International Symposium on Information Theory, 2005. ISIT 2005., pp.2036–2039, 2005.
[2] J. Cheng, K. Kamoi, and Y. Watanabe, "User identification by signature code for noisy multiple-access adder channel," 2006 IEEE International Symposium on Information Theory, pp.1974–1977, 2006.
[3] S. Lu, W. Hou, and J. Cheng, "A family of $(k+1)$-ary signature codes for noisy multiple-access adder channel," IEEE Trans. Inf. Theory, vol.61, no.11, pp.5848–5853, 2015.
[4] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," Commun. Pur. Appl. Math., vol.57, no.11, pp.1413–1457, 2004.
[5] S.J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale ℓ_1-regularized least squares," IEEE J. Sel. Top. Signal Process., vol.1, no.4, pp.606–617, 2007.
[6] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," Proc. 27th Asilomar Conference on Signals, Systems and Computers, pp.40–44, 1993.
[7] J.A. Tropp and A.C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," IEEE Trans. Inf. Theory, vol.53, no.12, pp.4655–4666, 2007.
[8] L. Liu, E.G. Larsson, W. Yu, P. Popovski, C. Stefanovic, and E. de Carvalho, "Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the internet of things," IEEE Signal Process. Mag., vol.35, no.5, pp.88–99, 2018.
[9] T. Hara and K. Ishibashi, "Grant-free non-orthogonal multiple access with multiple-antenna base station and its efficient receiver design," IEEE Access, vol.7, pp.175717–175726, 2019.
[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.
[11] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," 2016 54th Annual Allerton Conference

on Communication, Control, and Computing (Allerton), pp.341–346, 2016.
[12] M. Kim, N.I. Kim, W. Lee, and D.H. Cho, "Deep learning-aided SCMA," IEEE Commun. Lett., vol.22, no.4, pp.720–723, 2018.
[13] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1828–1837, 2018.
[14] D. Ito, S. Takabe, and T. Wadayama, "Trainable ISTA for sparse signal recovery," IEEE Trans. Signal Process., vol.67, no.12, pp.3113–3125, 2019.
[15] W. Kim, Y. Ahn, and B. Shim, "Deep neural network-based active user detection for grant-free noma systems," IEEE Trans. Commun., vol.68, no.4, pp.2143–2155, 2020.
[16] L. Wei, S. Lu, H. Kamabe, and J. Cheng, "User identification and channel estimation by DNN-based decoder on multiple-access channel," GLOBECOM 2020-2020 IEEE Global Communications Conference, pp.1–6, 2020.
[17] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," Proc. Fourteenth International Conference on Artificial Intelligence and Statistics, pp.315–323, 2011.
[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.
[19] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?," Advances in Neural Information Processing Systems, pp.2483–2493, 2018.
[20] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[21] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural networks for machine learning, vol.4, no.2, pp.26–31, 2012.
[22] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," Proc. thirteenth International Conference on Artificial Intelligence and Statistics, pp.249–256, 2010.
[23] S.S. Chen, D.L. Donoho, and M.A. Saunders, "Atomic decomposition by basis pursuit," SIAM Rev., vol.43, no.1, pp.129–159, 2001.
[24] M. Grant, S. Boyd, and Y. Ye, "Disciplined convex programming," in Global optimization, pp.155–210, Springer, 2006.

Lantian Wei was born in Hubei, China in 1995. He received B.E. and M.E. degrees from Dalian University, China in 2017 and Gifu University, Japan in 2020, respectively. He is currently working towards a Ph.D. at Gifu University. His research interests include wireless communication and machine learning.

Shan Lu received the B.S. and M.S. degrees in telecommunications engineering from Xidian University, Xi'an, China, in 2007 and 2010, respectively, and the Ph.D. degree in information and computer science from Doshisha University, Kyoto, Japan, in 2014. From 2014 to 2016, she was a research assistant at Doshisha University. Currently, she is an assistant professor in the Department of Electrical, Electronic and Computer Engineering, Gifu University. Her research interests are in the areas of multiuser coding, coding for nonvolatile memories, and communications theory.

Hiroshi Kamabe received the B.E. and M.E. degrees from Toyohashi University of Technology in 1982 and 1924, respectively. He received the Ph.D. from Nagoya University in 1996. He joined Mie University in 1982 and Gifu University in 1998. Since 2010, he has been a professor at Gifu University. He is now a vice dean of faculty of engineering. He served as an associate editor of IEICE Trans. Fundamentals (EA), an editor of Fundamentals Review, a chair of IEICE technical committee of Information Theory, the chief editor of IEICE Trans. Fundamentals (Japanese Edition) and a chair of TPC of ISITA2020, and so on. He is interested in constrained coding.

Jun Cheng received the B.S. and M.S. degrees in telecommunications engineering from Xidian University, Xi'an, China, in 1984 and 1987, respectively, and the Ph.D. degree in electrical engineering from Doshisha University, Kyoto, Japan, in 2000. From 1987 to 1994, he was an Assistant Professor and Lecturer in the Department of Information Engineering, Xidian University. From 1995 to 1996, he was an Associate Professor in the National Key Laboratory on Integrated Service Network, Xidian University. In April 2000, he joined ATR Adaptive Communications Research Laboratories, Kyoto, Japan, where he was a Visiting Researcher. From August 2002 to June 2003, he was working as a Staff Engineer at the R&D Center, Panasonic Mobile Communications Co., Ltd. (formerly Wireless Solution Labs., Matsushita Communication Industrial Co., Ltd.), Yokosuka, Japan. From July 2003 to March 2004, he was a Staff Engineer at Next-Generation Mobile Communications Development Center, Matsushita Electric Industrial Co., Ltd., Yokosuka, Japan. In April 2004, he joined Doshisha University, Kyoto, Japan. Currently, he is a Professor in the Department of Intelligent Information Engineering and Sciences, Doshisha University, Kyoto, Japan. From October 2016 to September 2017, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, Texas A&M University, USA. His research interests are in the areas of communications theory, information theory, coding theory, array signal processing, and radio communication systems.